

# **DAFTAR ISI**

DAFTAR ISI .....	i
DAFTAR GAMBAR .....	ii
DAFTAR TABEL .....	ii
BAB 1. PENGELOLAAN USER & GROUP PADA SISTEM OPERASI LINUX .....	4
BAB 2. PROCESS CONTROL .....	8
BAB 3. ACCESS CONTROL, ROOTLY POWER AND FILE SYSTEM - PART 1 ...	21
BAB 4. ACCESS CONTROL, ROOTLY POWER AND FILE SYSTEM - PART 2 ...	31
BAB 5. FILESYSTEM AND LOGICAL VOLUME MANAGER .....	41
BAB 6. FILESYSTEM AND LOGICAL VOLUME MANAGER – PART II .....	48
BAB 7. SAMBA DOMAIN CONTROLLER .....	59
BAB 8. PENGELOLAAN OBJEK PADA ACTIVE DIRECTORY MENGGUNAKAN POWERSHELL .....	65
BAB 9. MANAGING GROUP POLICY & ADVANCED AD MANAGEMENT WITH POWERSHELL .....	69
BAB 10. AUDITING & MONITORING .....	79

# DAFTAR GAMBAR

Gambar 1 Command Membuat Direktori ScrollRoyce .....	4
Gambar 2 Command Membuat User Baru .....	4
Gambar 3 Command Membuat Grup .....	5
Gambar 4 Memasukkan user ke dalam grup .....	5
Gambar 5 Membuat direktori data.....	5
Gambar 6 Membuat direktori per divisi .....	5
Gambar 7 Command menset permission pada Finance.....	5
Gambar 8 Menset Permission direktori HRD & GA.....	5
Gambar 9 Set Permission Direktori CEO .....	5
Gambar 10 Set Permission Direktori Legal .....	6
Gambar 11 Set Permission direktori Komisaris .....	6
Gambar 12 Output Command Ps .....	8
Gambar 13 Output Perintah sleep 1000 & .....	8
Gambar 14 Output perintah ps setelah menjalankan command sleep 1000 & .....	8
Gambar 15 Output Command ps -u.....	9
Gambar 16 Ouput Command ps -e .....	9
Gambar 17 Output command ps -f .....	9
Gambar 18 Output Command ps -ef.....	10
Gambar 19 Command membuay user baru .....	10
Gambar 20 Kode penjumlahan .....	11
Gambar 21 Kode pengurangan .....	11
Gambar 22 Kode perkalian .....	11
Gambar 23 Kode pembagian .....	12
Gambar 24 Menampilkan isi direktori tiap user .....	12
Gambar 25 Command menjalankan kode penjumlahan .....	12
Gambar 26 Command menjalankan kode pengurangan .....	12
Gambar 27 command menjalankan kode perkalian.....	13
Gambar 28 command menjalankan kode pembagian .....	13
Gambar 29 Hasil eksekusi kode perkalian dan pembagian .....	13
Gambar 30 Hasil eksekusi kode penjumlahan dan pengurangan .....	13
Gambar 31 Output command top .....	14

Gambar 32 Output command htop .....	14
Gambar 33 Kode prog_1.c.....	15
Gambar 34 gambar eksekusi prog_1.c.....	15
Gambar 35 menjalankan kembali command top .....	16
Gambar 36 mengecek nilai dari nice .....	16
Gambar 37 command renice .....	16
Gambar 38 mengecek perubahan nilai nice .....	17
Gambar 39 mengesekusi prog_1.c dengan nice .....	17
Gambar 40 membuat kode bahasa c baru .....	17
Gambar 41 menjalankan kode yang baru kita buat .....	18
Gambar 42 Output perintah ps -aux.....	18
Gambar 43 menterminasi proses yang sedang berjalan.....	18
Gambar 44 hasil terminasi proses.....	18
Gambar 45 membuat user baru.....	19
Gambar 46 mengeksekusi program pada user baru .....	19
Gambar 47 hasil eksekusi program pada user baru .....	19
Gambar 48 command menterminate proses pada user baru .....	19
Gambar 49 Output terminasi .....	19
Gambar 50 menjalankan program .....	20
Gambar 51 Output command strace .....	20
Gambar 52 penambahan file baru pada user.....	21
Gambar 53 membuat file crontab .....	21
Gambar 54 Menampilkan file crontab .....	21

# BAB 1. PENGELOLAAN USER & GROUP PADA SISTEM OPERASI LINUX

Soal:

Perusahaan Scroll Royce bergerak dalam bidang pertambangan, dengan Head Office yang berlokasi di Jakarta. Pada perusahaan terdapat beberapa divisi kerja, yaitu divisi Finance, HRD & GA, CEO, dan Legal yang dipimpin oleh seorang Komisaris perusahaan yaitu Bapak Binsar, yang mempunyai wewenang penuh untuk tiap divisi.

Pada Head Office telah diimplementasikan pembangunan jaringan lokal maupun internet, tetapi jaringan belum dikelola dengan baik, terutama dalam hal file sharing untuk masing-masing divisi dan penerapan pembatasan hak akses. Berdasarkan requirement dari pihak HRD, maka administrator jaringan diminta untuk membuat folder kerja dengan nama sesuai dengan masing-masing divisi, folder share informasi public, dan folder kerja untuk komisaris. Divisi yang pekerjaannya saling berhubungan dengan divisi lainnya akan mendapatkan hak akses terhadap folder kerja masing-masing divisi tersebut, dengan ketentuan sebagai berikut.

1. Folder Finance dapat diakses oleh bagian Finance, bagian CEO, dan juga komisaris dengan mendapatkan akses penuh.
2. Folder HRD & GA dapat diakses oleh bagian HRD & GA dan komisaris dengan hak akses penuh, sedangkan bagian Finance dan Legal dengan hak akses read dan execute.
3. Folder CEO hanya dapat diakses oleh CEO dan Komisaris dengan hak akses penuh.
4. Folder Legal dapat diakses oleh bagian Legal, CEO, dan Komisaris dengan hak akses penuh, serta dapat diakses oleh user Bylardo bagian Finance dengan hak akses read dan execute.
5. Folder Komisaris hanya dapat diakses oleh komisaris perusahaan.
6. Folder Info Publik dapat diakses oleh semua user.

Jawaban:

Membuat direktori /home/ScrollRoyce

```
[rioz@serveransible ~]$ sudo mkdir /home/ScrollRoyce
[sudo] password for rioxz:
[rioz@serveransible ~]$ █
```

Gambar 1 Command Membuat Direktori ScrollRoyce

Membuat user baru pada direktori /home/ScrollRoyce

```
[rioz@serveransible ~]$ sudo useradd Bylardo -d /home/ScrollRoyce
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
[root@serveransible ScrollRoyce]# sudo useradd Leo -d /home/ScrollRoyce
useradd: warning: the home directory already exists.
Not copying any file from skel directory into it.
[root@serveransible ScrollRoyce]# sudo passwd Leo
Changing password for user Leo.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ScrollRoyce]# █
```

Gambar 2 Command Membuat User Baru

Membuat group user

```
[root@serveransible data]# groupadd Finance
[root@serveransible data]# groupadd HRDGA
[root@serveransible data]# groupadd CEO
[root@serveransible data]# groupadd Legal
[root@serveransible data]# groupadd Komisaris
[root@serveransible data]#
```

Gambar 3 Command Membuat Grup

Menset user ke dalam grup

```
[root@serveransible data]# usermod -a -G Finance Bylardo | usermod -a -G Finance Leo
[root@serveransible data]# usermod -a -G HRDGA Sahat | usermod -a -G HRDGA Alexander
[root@serveransible data]# usermod -a -G CEO Christine | usermod -a -G CEO Christine
[root@serveransible data]# usermod -a -G Legal Ruhut | usermod -a -G Legal Togar
[root@serveransible data]# usermod -a -G Komisaris Binsar
[root@serveransible data]#
```

Gambar 4 Memasukkan user ke dalam grup

Kemudian mengcreate direktori /data/nama\_folder dan membuat direktori

```
[root@serveransible ~]# mkdir data
[root@serveransible ~]# ls data
[root@serveransible ~]# cd data
[root@serveransible data]#
```

Gambar 5 Membuat direktori data

```
[root@serveransible data]# mkdir Finance
[root@serveransible data]# mkdir CEO
[root@serveransible data]# mkdir Legal
[root@serveransible data]# mkdir Komisaris
[root@serveransible data]# mkdir Info
[root@serveransible data]# mkdir HRDGA
[root@serveransible data]# ls
CEO  Finance  HRDGA  Info  Komisaris  Legal
[root@serveransible data]#
```

Gambar 6 Membuat direktori per divisi

Kemudian kita melakukan set permission:

Folder Finance dapat diakses oleh bagian Finance, bagian CEO, dan juga komisaris dengan mendapatkan akses penuh.

```
[root@serveransible data]# setfacl -m g:Finance:rwx -R Finance
[root@serveransible data]# setfacl -m g:CEO:rwx -R Finance
[root@serveransible data]# setfacl -m g:Komisaris:rwx -R Finance
```

Gambar 7 Command menset permission pada Finance

Folder HRD & GA dapat diakses oleh bagian HRD & GA dan komisaris dengan hak akses penuh, sedangkan bagian Finance dan Legal dengan hak akses read dan execute.

```
[root@serveransible data]# setfacl -m g:HRDGA:rwx -R HRDGA
[root@serveransible data]# setfacl -m g:Komisaris:rwx -R HRDGA
[root@serveransible data]# setfacl -m g:Finance:rx -R HRDGA
[root@serveransible data]# setfacl -m g:Legal:rx -R HRDGA
[root@serveransible data]#
```

Gambar 8 Menset Permission direktori HRD & GA

Folder CEO hanya dapat diakses oleh CEO dan Komisaris dengan hak akses penuh.

```
[root@serveransible data]# setfacl -m g:CEO:rwx -R CEO
[root@serveransible data]# setfacl -m g:Komisaris:rwx -R CEO
```

Gambar 9 Set Permission Direktori CEO

Folder Legal dapat diakses oleh bagian Legal, CEO, dan Komisaris dengan hak akses penuh, serta dapat diakses oleh user Bylardo bagian Finance dengan hak akses read dan execute.

```
[root@serveransible data]# setfacl -m g:Legal:rwx -R Legal
[root@serveransible data]# setfacl -m g:CEO:rwx -R Legal
[root@serveransible data]# setfacl -m g:Komusaris:rwx -R Legal
setfacl: Option -m: Invalid argument near character 3
[root@serveransible data]# setfacl -m g:Komisaris:rwx -R Legal
[root@serveransible data]# setfacl -m u:Bylardo:rx -R Legal
[root@serveransible data]# █
```

Gambar 10 Set Permission Direktori Legal

Folder Komisaris hanya dapat diakses oleh komisaris perusahaan.

```
[root@serveransible data]# setfacl -m g:Komisaris:rwx -R Komisaris
```

Gambar 11 Set Permission direktori Komisaris

Script:

```
sudo useradd Bylardo -d /home/ScrollRoyce
ls
sudo passwd Bylardo
ls
sudo passwd Leo
sudo useradd Sahat -d /home/ScrollRoyce
sudo passwd Sahat
sudo useradd Alexander -d /home/ScrollRoyce
sudo passwd Alexander
sudo adduser Christine -d /home/ScrollRoyce
sudo passwd Christine
sudo adduser Ruth -d /home/ScrollRoyce
sudo passwd Ruth
sudo adduser Ruhut -d /home/ScrollRoyce
sudo passwd Ruhut
sudo adduser Togar -d /home/ScrollRoyce
sudo passwd Togar
sudo adduser Binsar -d /home/ScrollRoyce
sudo passwd Binsar
cd /data
cd
ls
mkdir data
ls data
cd data
mkdir Finance
mkdir HRD & GA
mkdir HRD_&_GA
mkdir HRD&GA
mkdir hrd&ga
rmdir Finance
ls
rmdir nrn
rm HRD
rmdir HRD
rmdir HRD_
rmdir hrd
```

```
mkdir Finance
mkdir CEO
mkdir Legal
mkdir Komisaris
mkdir Info
mkdir HRDGA
ls
groupadd Finance
groupadd HRDGA
groupadd CEO
groupadd Legal
groupadd Komisaris
usermod -a -G Bylardo
usermod -a -G Bylardo Leo
usermod -a -G Finance Bylardo | usermod -a -G Finance Leo
usermod -a -G HRDGA Sahat | usermod -a -G HRDGA Alexander
usermod -a -G CEO Christine | usermod -a -G CEO Christine
usermod -a -G Legal Ruhut | usermod -a -G Legal Togar
usermod -a -G Komisaris Binsar
setfacl -m g:Finance:rwx -R Finance
setfacl -m g:CEO:rwx -R Finance
setfacl -m g:Komisaris:rwx -R Finance
setfacl -m g:HRDGA:rwx -R HRDGA
setfacl -m g:Komisaris:rwx -R HRDGA
setfacl -m g:Finance:rx -R HRDGA
setfacl -m g:Legal:rx -R HRDGA
setfacl -m g:CEO:rwx -R CEO
setfacl -m g:Komisaris:rwx -R CEO
setfacl -m g:Legal:rwx -R Legal
setfacl -m g:CEO:rwx -R Legal
setfacl -m g:Komusaris:rwx -R Legal
setfacl -m g:Komisaris:rwx -R Legal
setfacl -m u:Bylardo:rx -R Legal
setfacl -m g:Komisaris:rwx -R Komusaris
setfacl -m g:Komisaris:rwx -R Komisaris
chmod 777 info
sudo chmod 777 info
```

## BAB 2. PROCESS CONTROL

### A. Tugas

#### 1. Monitoring dan Pengaturan Proses

Proses mana yang berjalan pada sistem saya?

- a. Perintah ps

- i. Buka terminal dan jalankan perintah ps, tampilkan hasilnya.

```
[root@serveransible ~]# ps
  PID TTY      TIME CMD
 9327 pts/0    00:00:00 bash
 9420 pts/0    00:00:00 ps
```

Gambar 12 Output Command Ps

- ii. Buka terminal lain dan jalankan perintah sleep 1000 &, yang akan memulai proses tidur (sleeping process) pada proses background. Kemudian jalankan perintah ps dan tampilkan hasilnya.

```
File Edit View Search Terminal Help
[root@serveransible ~]# sleep 1000 &
[1] 9498
[root@serveransible ~]#
```

Gambar 13 Output Perintah sleep 1000 &

- iii. Pada terminal pertama jalankan perintah ps lagi. Ada yang perlu diperhatikan bahwa proses sleep tidak muncul, meskipun waktu 1000 detik belum kadaluwarsa.

```
[root@serveransible ~]# ps
  PID TTY      TIME CMD
 9327 pts/0    00:00:00 bash
 9510 pts/0    00:00:00 ps
[root@serveransible ~]#
```

Gambar 14 Output perintah ps setelah menjalankan command sleep 1000 &

- iv. (Pertanyaan 1) Menurut anda mengapa terjadi demikian? (petunjuk: kolom TTY). Jawab: ps menampilkan aktif proses yang sedang berjalan dan pada output ps, terdapat kolom tty yang dimana tty menampilkan nama file terminal yang terhubung ke input standar.
  - v. Untuk menampilkan seluruh proses pada terminal pertama dapat dijalankan dengan perintah ps -u. Perhatikan kolom PID, setiap proses memiliki ID yang unik diberikan oleh kernel. Sleep 1000 kurang tidak begitu berguna, sehingga perlu untuk diakhiri dengan cara mengirimkan signal ke proses sesuai dengan PID yang dituju.

```
[root@serveransible ~]# ps -u
USER          PID %CPU %MEM    VSZ   RSS TTY      STAT START  TIME COMMAND
root        1648  0.7  1.1 385180 46864 tty1      Ssl+ 15:46  0:09 /usr/bin/X :0
root        9327  0.0  0.0 116896  3472 pts/0      Ss   15:49  0:00 bash
root        9449  0.0  0.0 116896  3476 pts/1      Ss+ 15:50  0:00 bash
root        9595  0.0  0.0 116900  3476 pts/0      S   15:55  0:00 bash
root        9821  0.0  0.0 155452 1880 pts/0      R+  16:07  0:00 ps -u
[root@serveransible ~]#
```

**Gambar 15 Output Command ps -u**

- vi. (Pertanyaan 2) Dalam kasus ini bagaimanakah cara untuk mengakhiri proses sleep tersebut? (petunjuk: signal dan PID).  
Jawab: dengan menggunakan command kill dan mengetahui PID dari proses yang akan diakhiri
- vii. (Pertanyaan 3) Selain menggunakan perintah ps secara umum, perintah lain yang dapat digunakan adalah ps -e / ps -f / ps -ef. Tampilkan dan beri pembahasan dari hasil ketiga perintah tersebut.

- ps -e

```
9188 ?      00:00:00 tracker-miner-a
9240 ?      00:00:00 evolution-calen
9258 ?      00:00:00 evolution-addre
9272 ?      00:00:00 evolution-addre
9279 ?      00:00:00 gvfsd-burn
9296 ?      00:00:00 ibus-engine-sim
9302 ?      00:00:00 fwupd
9305 ?      00:00:00 gvfsd-metadata
9320 ?      00:00:02 gnome-terminal-
9326 ?      00:00:00 gnome-pty-helpe
9327 pts/0    00:00:00 bash
9449 pts/1    00:00:00 bash
9531 ?      00:00:00 dhclient
9595 pts/0    00:00:00 bash
9715 ?      00:00:00 anacron
9775 ?      00:00:00 kworker/1:1
9786 ?      00:00:00 kworker/3:2
9867 ?      00:00:00 kworker/3:0
9892 ?      00:00:00 sleep
9893 ?      00:00:00 fprintd
9903 ?      00:00:00 kworker/3:3
9904 ?      00:00:00 systemd-hostnam
9912 pts/0    00:00:00 ps
```

**Gambar 16 Ouput Command ps -e**

Melihat proses yang tidak terkait dengan terminal

- Ps -f

```
[root@serveransible ~]# ps -f
UID          PID  PPID  C STIME TTY          TIME CMD
root        9327  9320  0 15:49 pts/0      00:00:00 bash
root        9595  9327  0 15:55 pts/0      00:00:00 bash
root        9959  9595  0 16:17 pts/0      00:00:00 ps -f
```

**Gambar 17 Output command ps -f**

Menampilkan proses berdasarkan field tertentu

- ps -ef

File	Edit	View	Search	Terminal	Help	
oot	9177	1	0	15:49 ?	00:00:00	/usr/libexec/evolution-calenda
oot	9178	8666	0	15:49 ?	00:00:00	/usr/libexec/tracker-extract
oot	9188	8666	0	15:49 ?	00:00:00	/usr/libexec/tracker-miner-app
oot	9240	9177	0	15:49 ?	00:00:00	/usr/libexec/evolution-calenda
oot	9258	1	0	15:49 ?	00:00:00	/usr/libexec/evolution-address
oot	9272	9258	0	15:49 ?	00:00:00	/usr/libexec/evolution-address
oot	9279	8716	0	15:49 ?	00:00:00	/usr/libexec/gvfsd-burn --spav
oot	9296	8887	0	15:49 ?	00:00:00	/usr/libexec/ibus-engine-simpl
oot	9302	1	0	15:49 ?	00:00:00	/usr/libexec/fwupd/fwupd
oot	9305	1	0	15:49 ?	00:00:00	/usr/libexec/gvfsd-metadata
oot	9320	1	0	15:49 ?	00:00:03	/usr/libexec/gnome-terminal-se
oot	9326	9320	0	15:49 ?	00:00:00	gnome-pty-helper
oot	9327	9320	0	15:49 pts/0	00:00:00	bash
oot	9449	9320	0	15:50 pts/1	00:00:00	bash
oot	9531	868	0	15:52 ?	00:00:00	/sbin/dhclient -d -q -sf /usr/
oot	9595	9327	0	15:55 pts/0	00:00:00	bash
oot	9715	1	0	16:01 ?	00:00:00	/usr/sbin/anacron -s
oot	9775	2	0	16:05 ?	00:00:00	[kworker/1:1]
oot	9786	2	0	16:05 ?	00:00:00	[kworker/3:2]
oot	9867	2	0	16:10 ?	00:00:00	[kworker/3:0]
oot	9903	2	0	16:13 ?	00:00:00	[kworker/3:3]
oot	10006	801	0	16:19 ?	00:00:00	sleep 60
oot	10014	9595	0	_16:19 pts/0	00:00:00	ps -ef

Gambar 18 Output Command ps -ef

Menampilkan proses yang sedang berjalan saat ini

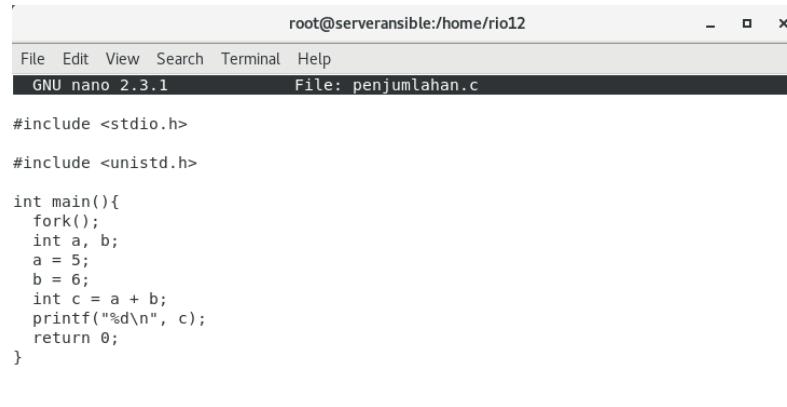
- viii. Buatlah dua pengguna pada sistem anda (anda bebas menentukan nama pengguna).

```
[root@serveransible ~]# useradd rio12
[root@serveransible ~]# passwd rio12
Changing password for user rio12.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd rio38
[root@serveransible ~]# passwd rio38
Changing password for user rio38.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

Gambar 19 Command membuat user baru

- b. Perintah top dan htop
- Gunakanlah dua pengguna yang telah anda tambahkan di sistem pada bagian a.
  - Berdasarkan pengetahuan anda pada matakuliah Sistem Operasi dan Sistem Programming, anda diharapkan untuk membuat program pengelolaan proses menggunakan system call fork().
  - Untuk masing-masing pengguna buatlah 2 program yang menggunakan system call fork() dengan ketentuan sebagai berikut:
    - Program untuk pengguna pertama adalah pertambahan dan pengurangan.

Penjumlahan



```
root@serveransible:/home/rio12
File Edit View Search Terminal Help
GNU nano 2.3.1          File: penjumlahan.c

#include <stdio.h>
#include <unistd.h>

int main(){
    fork();
    int a, b;
    a = 5;
    b = 6;
    int c = a + b;
    printf("%d\n", c);
    return 0;
}
```

Gambar 20 Kode penjumlahan

### Pengurangan



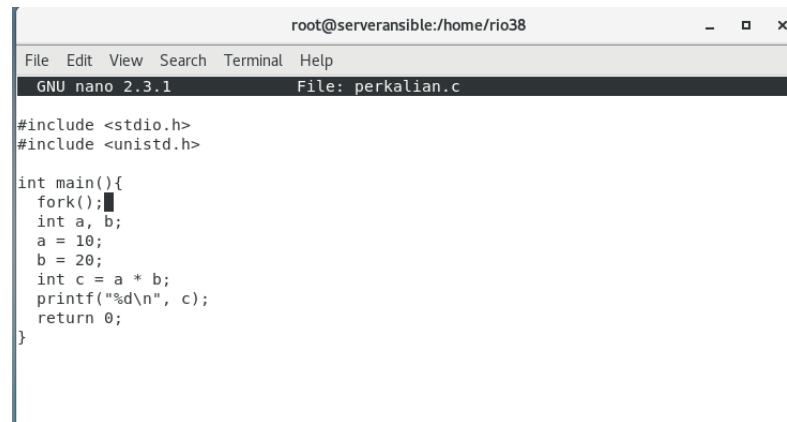
```
root@serveransible:/home/rio12
File Edit View Search Terminal Help
GNU nano 2.3.1          File: pengurangan.c

#include <stdio.h>
#include <unistd.h>

int main(){
    fork();
    int a , b;
    a = 6;
    b = 3;
    int c = a - b;
    printf("%d\n", c);
    return 0;
}
```

Gambar 21 Kode pengurangan

Program untuk pengguna kedua adalah perkalian dan pembagian:  
Perkalian



```
root@serveransible:/home/rio38
File Edit View Search Terminal Help
GNU nano 2.3.1          File: perkalian.c

#include <stdio.h>
#include <unistd.h>

int main(){
    fork();■
    int a, b;
    a = 10;
    b = 20;
    int c = a * b;
    printf("%d\n", c);
    return 0;
}
```

Gambar 22 Kode perkalian

### Pembagian

```
root@serveransible:/home/rio38
File Edit View Search Terminal Help
GNU nano 2.3.1          File: pembagian.c

#include <stdio.h>
#include <unistd.h>

int main(){
    fork();
    int a, b;
    a = 200;
    b = 10;
    int c = a / b;
    printf("%d\n", c);
    return 0;
}
```

Gambar 23 Kode pembagian

Program diletakkan di home folder masing-masing pengguna

```
[root@serveransible rio38]# ls
pembagian.c  perkalian.c
[root@serveransible rio38]#
[root@serveransible rio12]# ls
pengurangan.c  penjumlahan.c
```

Gambar 24 Menampilkan isi direktori tiap user

- iv. Keempat program tersebut dibuatkan menjadi program zombie.  
Penjumlahan.c

```
[root@serveransible rio12]# gcc penjumlahan.c -o penjumlahan
[root@serveransible rio12]# ./penjumlahan
11
11
[root@serveransible rio12]#
```

Gambar 25 Command menjalankan kode penjumlahan

Pengurangan.c

```
[root@serveransible rio12]# gcc pengurangan.c -o pengurangan
[root@serveransible rio12]# ./pengurangan
3
3
[root@serveransible rio12]#
```

Gambar 26 Command menjalankan kode pengurangan

Perkalian.c

```
[root@serveransible rio38]# gcc perkalian.c -o perkalian
[root@serveransible rio38]# ./perkalian
200
200
[root@serveransible rio38]#
```

Gambar 27 command menjalankan kode perkalian

#### Pembagian.c

```
[root@serveransible rio38]# gcc pembagian.c -o pembagian
[root@serveransible rio38]# ./pembagian
20
20
[root@serveransible rio38]#
```

Gambar 28 command menjalankan kode pembagian

- i. Jalankan program yang telah ditentukan ke setiap pengguna di terminal masing-masing pengguna.

Pada terminal rio38:

```
root@serveransible:/home/rio38
File Edit View Search Terminal Help
[root@serveransible ~]# cd /home/rio38
[root@serveransible rio38]# gcc -o pembagian pembagian.c
[root@serveransible rio38]# ./pembagian
20
20
[root@serveransible rio38]# gcc -o perkalian perkalian.c
[root@serveransible rio38]# ./perkalian
200
200
[root@serveransible rio38]#
```

Gambar 29 Hasil eksekusi kode perkalian dan pembagian

Pada terminal rio12:

```
[root@serveransible rio38]# cd /home/rio12
[root@serveransible rio12]# ls
9000_pengurangan pengurangan.c penjumlahan penjumlahan.c
[root@serveransible rio12]# ./penjumlahan
11
11
[root@serveransible rio12]# ./pengurangan
3
3
[root@serveransible rio12]#
```

Gambar 30 Hasil eksekusi kode penjumlahan dan pengurangan

- ii. Pada terminal utama jalankan perintah top, tampilkan dan berikan pembahasan hasilnya.

top - 14:25:56 up 24 min, 4 users, load average: 0.01, 0.03, 0.05										
Tasks: 306 total, 1 running, 305 sleeping, 0 stopped, 0 zombie										
%Cpu(s): 0.0 us, 0.0 sy, 0.0 ni, 99.9 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st										
KiB Mem : 4158224 total, 1929764 free, 935900 used, 1292560 buff/cache										
KiB Swap: 2097148 total, 2097148 free, 0 used. 2962172 avail Mem										
PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+ COMMAND
8829	root	20	0	3817976	176592	69032	S	0.7	4.2	1:02.08 gnome-she+
2118	root	20	0	391060	52500	25556	S	0.3	1.3	0:19.30 X
9131	root	20	0	610092	27012	18912	S	0.3	0.6	0:03.62 vmtoolsd
9300	root	20	0	738536	29280	16792	S	0.3	0.7	0:10.33 gnome-ter+
1	root	20	0	191284	4316	2612	S	0.0	0.1	0:03.69 systemd
2	root	20	0	0	0	0	S	0.0	0.0	0:00.04 kthreadd
4	root	0	-20	0	0	0	S	0.0	0.0	0:00.00 kworker/0+
6	root	20	0	0	0	0	S	0.0	0.0	0:00.11 ksftirqd+
7	root	rt	0	0	0	0	S	0.0	0.0	0:00.73 migration+
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00 rcu_bh
9	root	20	0	0	0	0	S	0.0	0.0	0:01.94 rcu_sched
10	root	0	-20	0	0	0	S	0.0	0.0	0:00.00 lru-add-d+
11	root	rt	0	0	0	0	S	0.0	0.0	0:00.16 watchdog/0
12	root	rt	0	0	0	0	S	0.0	0.0	0:00.01 watchdog/1
13	root	rt	0	0	0	0	S	0.0	0.0	0:00.82 migration+
14	root	20	0	0	0	0	S	0.0	0.0	0:00.04 ksftirqd+
16	root	0	-20	0	0	0	S	0.0	0.0	0:00.00 kworker/l1

Gambar 31 Output command top

Pembahasan: terdapat 306 total task, terdapat 1 proses yang berjalan dan 305 dari total process sedang sleeping.

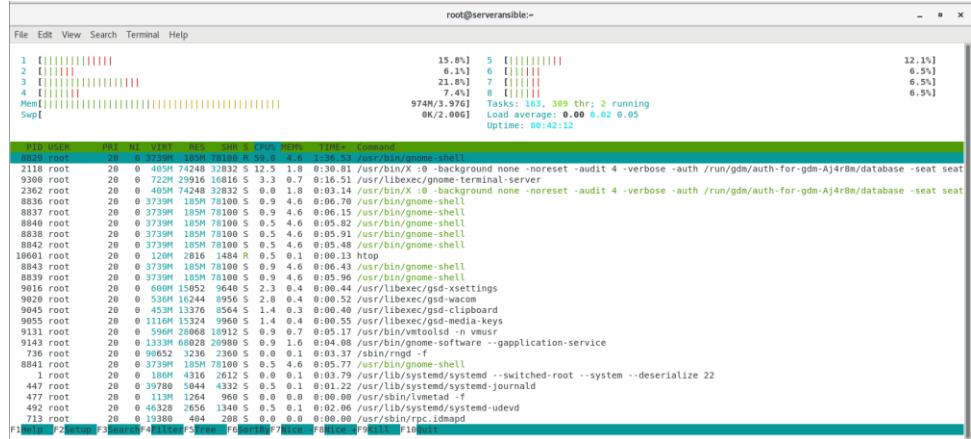
- iii. (Pertanyaan 1) Dalam kasus ini bagaimanakah cara untuk mengakhiri proses zombie tersebut?

Jawab: dengan menggunakan perintah kill namun kita harus mengetahui pid dari proses tersebut.

- iv. (Pertanyaan 2) Apakah anda menemukan perbedaan antara perintah ps dengan perintah top? Berikan penjelasan anda.

Jawab: ps menampilkan proses yang sedang berjalan sedangkan top menampilkan seluruh proses yang sedang berjalan, terminated, ataupun sedang sleep beserta dengan persentase penggunaan CPU dan memori

- v. Selain top terdapat perintah htop (open source software yang harus diinstall).



Gambar 32 Output command htop

- vi. Jalankan kembali program zombie yang telah anda ciptakan, kemudian jalankan perintah htop dan ps –ef.

- vii. (Pertanyaan 3) Apakah perbedaan dari kedua perintah tersebut?

Jawab: htop berfungsi untuk memantau proses system dan mengelola proses. Sedangkan ps -ef hanya menampilkan proses yang sedang berjalan saat ini

c. Perintah nice dan renice

- i. Buatlah sebuah program sederhana dengan nama prog\_1.c, yang bertujuan untuk mencetak nilai dari 1 sampai dengan 50 dengan waktu sleep sepanjang 5 detik setiap 1 iterasi selesai. Jalankan program sehingga menghasilkan luaran seperti gambar berikut.

Kode dalam prog\_1.c

The screenshot shows a terminal window titled 'root@serveransible:/home/rio12'. The window title bar also displays 'File Edit View Search Terminal Help' and 'GNU nano 2.3.1'. Below the title bar, the file name 'prog\_1.c' is shown in the status bar. The main area of the terminal contains the following C code:

```
#include<stdio.h>
#include<unistd.h>
int main(){
    int a;
    for(a = 1; a <= 50; a++){
        printf("%d\n",a);
        if(a == 50){
            a = 1;
            sleep(5);
        }
    }
    return 0;
}
```

Gambar 33 Kode prog\_1.c

Hasil eksekusi:

The screenshot shows a terminal window with the following command history and output:

```
[root@serveransible rio12]# gcc -o prog_1 prog_1.c
[root@serveransible rio12]# ./prog_1.c
bash: ./prog_1.c: Permission denied
[root@serveransible rio12]# ./prog_1
1
2
3
4
5
6
7
8
9
10
11
12
```

The terminal window has a green header bar with the text '31 root 20 0 596M 27392 18960 5 0.0 0.0 0.0 / 0:07.73'.

Gambar 34 gambar eksekusi prog\_1.c

- ii. Buka terminal yang baru, kemudian jalankan program top untuk melihat daftar proses yang sedang berjalan. Identifikasi nilai nice untuk proses prog\_1.

Gambar 35 menjalankan kembali command top

- iii. (Pertanyaan 1) Berapakah nilai nice? Mengapa demikian? Tampilkan hasil observasi anda.

Jawab: nilai nice nya adalah 0, yang dimana rentang nilai nice adalah 0 – 40

```
[root@serveransible ~]# top
top - 15:55:01 up 1:53, 4 users, load average: 0.00, 0.01, 0.05
Tasks: 320 total, 1 running, 305 sleeping, 14 stopped, 0 zombie
%Cpu0 : 0.7 us, 0.3 sy, 0.0 ni, 99.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
%Cpu1 : 0.3 us, 0.3 sy, 0.0 ni, 99.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
%Cpu2 : 0.3 us, 0.3 sy, 0.0 ni, 99.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
%Cpu3 : 0.3 us, 0.3 sy, 0.0 ni, 99.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
%Cpu4 : 0.7 us, 0.7 sy, 0.0 ni, 98.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
%Cpu5 : 0.0 us, 0.3 sy, 0.0 ni, 99.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
%Cpu6 : 0.7 us, 1.0 sy, 0.0 ni, 98.3 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
%Cpu7 : 0.3 us, 1.0 sy, 0.0 ni, 98.7 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 4158224 total, 1638708 free, 1109896 used, 1409620 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used. 2735128 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
8829	root	20	0	3913540	198976	70128	S	8.6	4.8	3:40.67	gnome-shell
2118	root	20	0	426464	87148	42972	S	3.3	2.1	0:59.89	X
9300	root	20	0	750996	32304	17608	S	1.7	0.8	0:30.53	gnome-terminal-
11733	root	20	0	162236	2452	1584	S	0.7	0.1	0:00.41	top
9	root	20	0	0	0	0	S	0.3	0.0	0:05.20	rcu_sched
748	root	20	0	360916	5244	3980	S	0.3	0.1	0:11.28	vmtoolsd
1881	root	20	0	0	0	0	S	0.3	0.0	0:05.50	kworker/7:3
9131	root	20	0	610664	27492	18960	S	0.3	0.7	0:12.12	vmtoolsd
11828	root	20	0	162236	2480	1584	R	0.3	0.1	0:00.18	top
1	root	20	0	191284	4316	2612	S	0.0	0.1	0:04.21	systemd

Gambar 36 mengecek nilai dari nice

- iv. (Pertanyaan 2) Berapakah nilai nice? Mengapa demikian? Tampilkan hasil observasi anda.

Jawab: nilai nice tetap 0

- v. Buka terminal yang baru, kemudian dalam keadaan proses prog\_1 masih berjalan, ubahlah nilai prioriti proses menggunakan perintah renice dengan nilai -20. Kemudian, perhatikan kembali program top yang sedang berjalan.

```
[root@serveransible rio12]# renice '-20' -p '8829'
8829 (process ID) old priority 0, new priority -20
[root@serveransible rio12]# █
```

Gambar 37 command renice

Top:

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
8829	root	0	-20	3913540	198992	70132	S	12.3	4.8	3:51.70	gnome-shell

- vi. (Pertanyaan 3) berapakah nilai nice? Mengapa demikian? Tampilkan hasil observasi anda.

Jawab: nilai nice nya adalah 0 dikarenakan sebelumnya kita telah mengubah nilai nice yang sebelumnya 20 menjadi -20

```

root@serveransible:/home/rio55
File Edit View Search Terminal Help
top - 15:27:30 up 1:26, 4 users, load average: 0.00, 0.02, 0.05
Tasks: 315 total, 2 running, 305 sleeping, 8 stopped, 0 zombie
%Cpu(s): 0.3 us, 0.7 sy, 0.0 ni, 99.0 id, 0.0 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem : 4158224 total, 1847288 free, 994376 used, 1316560 buff/cache
KiB Swap: 2097148 total, 2097148 free, 0 used. 2881852 avail Mem

PID USER      PR  NI    VIRT    RES    SHR S %CPU %MEM     TIME+ COMMAND
8829 root      20   0 3920836 206100 77424 S  7.7  5.0  3:04.02 gnome-she+
2118 root      20   0 431036 89344 45168 S  4.1  2.1  0:50.01 X
9300 root      20   0 750996 32096 17608 S  2.0  0.8  0:25.35 gnome-ter+
1 root       20   0 191284 4316 2612 S  0.0  0.1  0:04.10 systemd
2 root       20   0     0     0     0 S  0.0  0.0  0:00.04 kthreadd
4 root      0 -20     0     0     0 S  0.0  0.0  0:00.00 kworker/0+

```

Gambar 38 mengecek perubahan nilai nice

- vii. Hentikan proses dari prog\_1. Kemudian jalankan program prog\_1 kembali namun dengan menggunakan perintah nice dengan parameter prioriti 15.

```

[root@serveransible rio12]# sudo nice -n -15 gcc -o prog_1 prog_1.c
[root@serveransible rio12]#

```

Gambar 39 mengesekusi prog\_1.c dengan nice

- viii. Jalankan program top, kemudian identifikasi nilai nice untuk process prog\_1.

#### d. Perintah Signal

- i. Jalankanlah 3 program editor vi pada terminal yang berbeda. Kemudian buka terminal baru untuk menterminasi ketiga program tersebut.

Nama file: 1.c , 2.c, 3.c

```

#include<cs51.h>
#include<unistd.h>
int main(){
    int a;
    for(a = 1; a <= 10; a++){
        printf("%d\n", a);
        if(a == 10){
            a = 1;
            sleep(2);
        }
    }
    return 0;
}

```

Gambar 40 membuat kode bahasa c baru

- ii. Sebelum menterminasi ketiga program tampilkan terlebih dahulu bahwa ketiga program tersebut adalah proses yang sedang berjalan pada sistem anda.

```

root@serveransible:~# ps
 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
root@serveransible:~# ps
 21 22 23 24 25 26 27 28 29 30
root@serveransible:~# ps
 31 32 33 34 35 36 37 38 39 40

```

Gambar 41 menjalankan kode yang baru kita buat

- iii. Buka terminal yang baru kemudian terminasi seluruh proses vi.  
Jawab: kita mengetik perintah ps -aux untuk melihat proses.

```

root      9698  0.0  0.0  4216   352 pts/0    S+   08:18   0:00 ./1
root      9710  0.0  0.0  4216   352 pts/1    S+   08:18   0:00 ./2
root      9730  0.0  0.0  4216   356 pts/2    S+   08:18   0:00 ./3

```

Gambar 42 Output perintah ps -aux

- iv. (Pertanyaan 1) Bagaimanakah cara untuk menterminasi seluruh proses vi?  
Tampilkan perintah yang anda gunakan dan hasil akhirnya.  
Jawab: dengan menggunakan command kill

```

[root@serveransible ~]# kill -9 9698 | kill -9 9710 | kill -9 9730
[root@serveransible ~]#

```

Gambar 43 menterminasi proses yang sedang berjalan

Hasil:

```

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 Killed
root@serveransible ~]# ps
21 22 23 24 25 26 27 28 29 30 Killed
root@serveransible ~]# ps
31 32 33 34 35 36 37 38 39 40 Killed

```

Gambar 44 hasil terminasi proses

- v. Buatlah 2 akun pengguna dengan nama user\_1 dan user\_2.  
vi. Jalankan 2 program apapun yang dieksekusi oleh masing-masing pengguna.  
vii. (Pertanyaan 2) Bagaimanakah caranya untuk menterminasi seluruh proses yang dijalankan oleh masing-masing pengguna (petunjuk: pkill). Tampilkan perintah

yang anda gunakan dan hasil akhirnya.

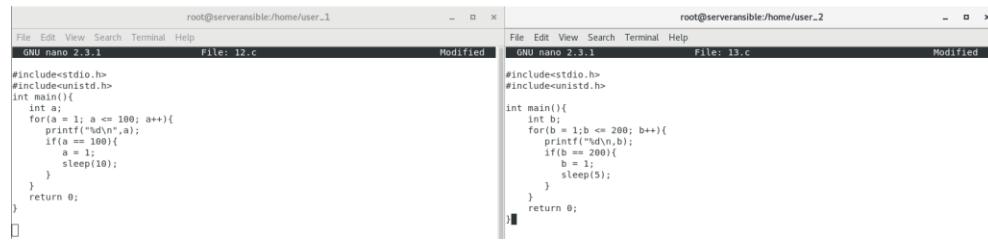
### Membuat user\_1 dan user\_2

```
[root@serveransible ~]# useradd user_1
[root@serveransible ~]# passwd user_1
Changing password for user user_1.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd user_2
[root@serveransible ~]# passwd user_2
Changing password for user user_2.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]#
```

Gambar 45 membuat user baru

Menjalankan program pada masing-masing user:

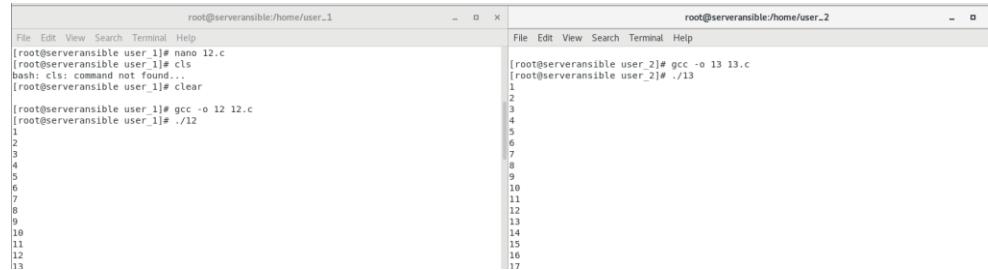
### Kode program:



```
root@serveransible:/home/user_1          root@serveransible:/home/user_2
File Edit View Search Terminal Help        File Edit View Search Terminal Help
GNU nano 2.3.1                         GNU nano 2.3.1
File: 12.c                               File: 13.c
Modified                                Modified
#include<stdio.h>
#include<unistd.h>
int main(){
    int a;
    for(a = 1; a <= 100; a++){
        printf("%d\n",a);
        if(a == 100){
            a = 1;
            sleep(10);
        }
    }
    return 0;
}
root@serveransible:/home/user_1          root@serveransible:/home/user_2
File Edit View Search Terminal Help
GNU nano 2.3.1
File: 13.c
Modified
#include<stdio.h>
#include<unistd.h>
int main(){
    int b;
    for(b = 1;b <= 200; b++){
        printf("%d\n",b);
        if(b == 200){
            b = 1;
            sleep(5);
        }
    }
    return 0;
}
```

Gambar 46 mengeksekusi program pada user baru

Hasil eksekusi:



```
root@serveransible:/home/user_1          root@serveransible:/home/user_2
File Edit View Search Terminal Help        File Edit View Search Terminal Help
root@serveransible user_1]# nano 12.c
[root@serveransible user_1]# cls
bash: cls: command not found...
[root@serveransible user_1]# clear
[root@serveransible user_1]# gcc -o 12 12.c
[root@serveransible user_1]# ./12
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
root@serveransible user_1]#          root@serveransible user_2]# gcc -o 13 13.c
[root@serveransible user_2]# ./13
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
```

Gambar 47 hasil eksekusi program pada user baru

Cara menterminasi proses kedua user:

```
[root@serveransible ~]# kill -9 10305 10306
[root@serveransible ~]#
```

Gambar 48 command menterminate proses pada user baru

Hasil:



```
97
98
99
100
Killed
[root@serveransible user_1]#
```

	197 198 199 200 Killed [root@serveransible user_2]#
--	--

Gambar 49 Output terminasi

## e. Perintah Strace

- i. Jalankan sebuah program misalnya prog\_1 yang telah anda buat sebelumnya.

```
[root@serveransible ~]# cd /home/rio12/  
[root@serveransible rio12]# ls  
pengurangan pengurangan.c penjumlahan penjumlahan.c prog_1 prog_1.c  
[root@serveransible rio12]# ./prog_1  
1  
2  
3  
4  
5  
6
```

Gambar 50 menjalankan program

- ii. Kemudian gunakan perintah strace untuk menggali informasi lebih lengkap dari proses yang anda eksekusi pada poin i. Tampilkan perintah yang anda gunakan dan hasil akhirnya, kemudian berikan penjelasan anda terhadap hasil dari perintah strace.

Hasil:

```
[root@serveransible rio12]# strace -p 10486  
strace: Process 10486 attached  
restart_syscall(<... resuming interrupted read ...>) = 0  
write(1, "2\n", 2) = 2  
write(1, "3\n", 2) = 2  
write(1, "4\n", 2) = 2  
write(1, "5\n", 2) = 2  
write(1, "6\n", 2) = 2  
write(1, "7\n", 2) = 2  
write(1, "8\n", 2) = 2  
write(1, "9\n", 2) = 2  
write(1, "10\n", 3) = 3
```

Gambar 51 Output command strace

Penjelasan: 10486 merupakan PID dari eksekusi prog\_1.c

```
[root    10486  0.0  0.0  4216  356 pts/1  S+  08:44  0:00 ./prog_1  
root    10498  0.2  0.0 116896 3480 pts/0  Ss  08:45  0:00 bash  
root    10512  0.0  0.1 350536  6796 ?      Sl  08:45  0:00 /usr/sbin/abrt-dbus  
root    10576  0.0  0.0 108056   356 ?      S   08:46  0:00 sleep 60  
root    10608  0.0  0.0 157536 1928 pts/0  R+  08:46  0:00 ps -aux
```

## 2. Automasi menggunakan cron

- Dari poin 1.b.iii., buatlah cron job yang akan menyalin berkas dari home folder ke folder lainnya (anda dapat menentukan lokasi dan nama foldernya). Tambahkan 2 berkas lainnya dihome folder masing-masing pengguna misalnya berkas .txt.

The image shows two terminal windows side-by-side. The left window, titled 'root@serveransible:/home/rio12', shows the command 'nano berkas.txt' being run, which creates a file containing 'berkas.txt' and 'pengurangan.c'. The right window, titled 'root@serveransible:/home/rio38', shows the command 'nano berkas.txt' being run, which creates a file containing 'berkas.txt' and 'pembagian.c'.

Gambar 52 penambahan file baru pada user

- b. File cron adalah unik untuk setiap pengguna dan hanya bisa dieksekusi oleh pengguna yang terkait saja dan juga root.
- c. Spesifikasi untuk cron-nya adalah cron dijalankan pada pukul 10:00 a.m. pada bulan 09, tahun 2021 setiap hari.

```
[root@serveransible ~]# crontab -l
50 10 12 9 1-7 cp -R /home/rio38 /home/rio12
```

Gambar 53 membuat file crontab

- d. Tampilkan seluruh proses, file crontab, dan juga hasil dari cron job.

```
[root@serveransible ~]# crontab -e
crontab: installing new crontab
[root@serveransible ~]# crontab -l
50 10 12 9 1-7 cp -R /home/rio38 /home/rio12
[root@serveransible ~]#
```

```
[root@serveransible cron]# cd /var/spool/cron
[root@serveransible cron]# ls
crontabs  root
[root@serveransible cron]#
```

Gambar 54 Menampilkan file crontab

## BAB 3. ACCESS CONTROL, ROOTLY POWER AND FILE SYSTEM - PART 1

Tugas

Pada skenario ini terdapat pengguna dan group pengguna sebagai berikut:

Group: Master Good Guys	Group: Master Bad Guys
Dumbledore	Voldemort
Group: Dosen Good Guys	Group: Dosen Bad Guys
Snape	Lucius
Sirius	Bellatrix
Group: Siswa Good Guys	Group: Siswa Bad Guys
Harry	Malfoy
Hermione	Seamus
	Squirrel

Sebagai kondisi awal, Anda diperbolehkan untuk menggunakan akun root untuk melakukan operasi berikut:

1. Menciptakan pengguna Dumbledore dan Voldemort.

Menciptakan pengguna baru dengan command adduser <nama user> dan menambahkan password untuk user baru dengan command passwd <nama user>

```
File Edit View Search Terminal Help
[root@serveransibile ~]# adduser Dumbledore
[root@serveransibile ~]# passwd Dumbledore
Changing password for user Dumbledore.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransibile ~]# adduser Voldemort
[root@serveransibile ~]# passwd Voldemort
passwd: Unknown user name 'Voldermort'.
[root@serveransibile ~]# passwd Voldemort
Changing password for user Voldemort.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransibile ~]#
```

Dan kemudian kita memasukkan user berdasarkan grupnya

Dengan membuat grup baru dengan command groupadd <nama Grup>

Dan menset user ke dalam grup dengan command usermod -a -G <nama grup> <nama user>

```
[root@serveransibile ~]# groupadd MasterGoodGuys
[root@serveransibile ~]# usermod -a -G MasterGoodGuys Dumbledore
[root@serveransibile ~]#
```

2. Mendaftarkan kedua pengguna menjadi superuser dengan wewenang penuh pada sistem setara dengan root.

Mendaftarkan kedua user menjadi superuser dengan mengetik masuk ke file /etc/sudoers dengan menggunakan command nano /etc/sudoers dan menambahkan user seperti gambar dibawah

```
## which machines (the sudoers file can be shared between multiple
## systems).
## Syntax:
##
##       user      MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)      ALL
Dumbledore  ALL=(ALL)    ALL
Voldemort   ALL=(ALL)    ALL
```

Setelah Dumbledore setara dengan root, maka dia dapat melakukan operasi berikut:

1. Dumbledore menciptakan pengguna pada group dosen yaitu Snape dan Sirius, lengkap dengan pengaturan password dan home directory dari kedua pengguna.  
Pertama kita membuat direktori baru untuk user yang akan kita buat dengan nama DosenGoodGuys

```
[Dumbledore@serveransible ~]$ sudo mkdir /home/DosenGoodGuys
```

```
We trust you have received the usual lecture from the local System
Administrator. It usually boils down to these three things:
```

```
#1) Respect the privacy of others.
#2) Think before you type.
#3) With great power comes great responsibility.
```

```
[sudo] password for Dumbledore:
[Dumbledore@serveransible ~]$ ls
```

Kemudian kita menambahkan user baru dan menset home directory

```
[Dumbledore@serveransible DosenGoodGuys]$ sudo adduser -m -d /home/Snape Snape
Creating mailbox file: File exists
[Dumbledore@serveransible DosenGoodGuys]$ sudo passwd Snape
Changing password for user Snape.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[Dumbledore@serveransible DosenGoodGuys]$ 
[Dumbledore@serveransible DosenGoodGuys]$ sudo adduser -m -d /home/Sirius Sirius
Creating mailbox file: File exists
[Dumbledore@serveransible DosenGoodGuys]$ passwd Sirius
passwd: Only root can specify a user name.
[Dumbledore@serveransible DosenGoodGuys]$ sudo passwd Sirius
Changing password for user Sirius.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[Dumbledore@serveransible DosenGoodGuys]$ 
```

2. Mengelompokkan Snape dan Sirius pada group dosen.

Kemudian kita menambahkan grup baru bernama DosenGoodGuys

```
[Dumbledore@serveransible ~]$ sudo groupadd DosenGoodGuys
```

Dan mengelompokkan user masuk ke grup

```
[Dumbledore@serveransible ~]$ sudo usermod -a -G DosenGoodGuys Snape | sudo usermod -a -G DosenGoodGuys Sirius
```

3. Memberikan wewenang kepada mereka menjadi superuser yang hanya dapat melakukan penambahan akun pengguna lainnya.

Masuk ke /etc/sudoers dan menset user menjadi superuser dengan pembatasan akses

```
## The COMMANDS section may have other options added to it.  
##  
## Allow root to run any commands anywhere  
root    ALL=(ALL)      ALL  
Dumbledore   ALL=(ALL)      ALL  
Voldemort    ALL=(ALL)      ALL  
Snape        ALL=/sbin/adduser, /bin/passwd, /sbin/usermod, /sbin/groupadd  
Sirius       ALL=/sbin/adduser, /bin/passwd, /sbin/usermod, /sbin/groupadd
```

Setelah akun pengguna Snape dan Sirius tercipta, maka mereka dapat melakukan operasi berikut:

1. Sirius menciptakan akun pengguna pada group siswa yaitu Harry dan Hermione, lengkap dengan pengaturan password dan home directory dari kedua pengguna.

Membuat user harry dan Hermione

```
[Sirius@serveransible ~]$ sudo useradd -m -d /home/Harry Harry  
useradd: warning: the home directory already exists.  
Not copying any file from skel directory into it.  
Creating mailbox file: File exists  
[Sirius@serveransible ~]$ sudo passwd Harry  
Changing password for user Harry.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[Sirius@serveransible ~]$ █
```

```
[Sirius@serveransible ~]$ sudo useradd -m -d /home/Hermione Hermione  
[sudo] password for Sirius:  
[Sirius@serveransible ~]$ sudo passwd Hermione  
Changing password for user Hermione.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[Sirius@serveransible ~]$
```

2. Snape mengelompokkan Harry dan Hermione ke dalam group siswa.

Kita membuat grup SiswaGoodGuys dengan command sudo groupadd <Nama grup>

```
[Snape@serveransible ~]$ sudo groupadd SiswaGoodGuys
```

We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:

- #1) Respect the privacy of others.
- #2) Think before you type.
- #3) With great power comes great responsibility.

```
[sudo] password for Snape:
```

```
[Snape@serveransible ~]$ █
```

Kemudian menset akun ke grup dengan command usermod -a -G <nama grup> <nama user>

```
[Snape@serveransible ~]$ sudo usermod -a -G SiswaGoodGuys Harry
```

```
[Snape@serveransible ~]$ sudo usermod -a -G SiswaGoodGuys Hermione
```

```
[Snape@serveransible ~]$
```

Pada sistem yang sama, Voldemort juga melakukan hal yang sama yaitu:

1. Menciptakan pengguna pada group dosen yaitu Lucius dan Bellatrix, lengkap dengan pengaturan password dan home directory dari kedua pengguna.

Kita membuat grup DosenBadGuys dengan command sudo groupadd DosenBadGuys

```
[Voldemort@serveransible ~]$ sudo groupadd DosenBadGuys
```

Kemudian kita membuat user baru

```
[Voldemort@serveransible ~]$ sudo useradd -m -d /home/Lucius Lucius
```

```
[Voldemort@serveransible ~]$ sudo passwd Lucius
```

```
Changing password for user Lucius.
```

```
New password:
```

```
Retype new password:
```

```
passwd: all authentication tokens updated successfully.
```

```
[Voldemort@serveransible ~]$ sudo useradd -m -d /home/Bellatrix Bellatrix
```

```
[Voldemort@serveransible ~]$ sudo passwd Bellatrix
```

```
Changing password for user Bellatrix.
```

```
New password:
```

```
Retype new password:
```

```
passwd: all authentication tokens updated successfully.
```

```
[Voldemort@serveransible ~]$
```

2. Mengelompokkan Lucius dan Bellatrix pada group dosen.

Kemudian kita mengelompokkan user yang telah dibuat tadi ke dalam grup yang sebelumnya telah dibuat

```
[Voldemort@serveransible ~]$ sudo usermod -a -G DosenBadGuys Lucius
```

```
[Voldemort@serveransible ~]$ sudo usermod -a -G DosenBadGuys Bellatrix
```

```
[Voldemort@serveransible ~]$
```

3. Memberikan wewenang kepada mereka menjadi superuser yang dapat menambah dan menghapus akun pengguna lainnya.

Dengan masuk kembali ke file /etc/sudoers

```
## The COMMANDS section may have other options added to it.  
##  
## Allow root to run any commands anywhere  
root    ALL=(ALL)      ALL  
Dumbledore  ALL=(ALL)      ALL  
Voldemort   ALL=(ALL)      ALL  
Snape      ALL=/sbin/adduser, /bin/passwd, /sbin/usermod, /sbin/groupadd  
Sirius      ALL=/sbin/adduser, /bin/passwd, /sbin/usermod, /sbin/groupadd  
Lucius      ALL=/sbin/adduser, /bin/passwd, /sbin/usermod, /sbin/groupadd  
Bellatrix   ALL=/sbin/adduser, /bin/passwd, /sbin/usermod, /sbin/groupadd
```

Kemudian Lucius, meminta Bellatrix untuk melakukan operasi berikut:

1. Menciptakan akun Malfoy dan Seamus, lengkap dengan pengaturan password dan home directory dari kedua pengguna.

Bellatrix membuat 2 akun baru

```
[Bellatrix@serveransible ~]$ sudo useradd -m -d /home/Malfoy Malfoy  
  
We trust you have received the usual lecture from the local System  
Administrator. It usually boils down to these three things:  
  
#1) Respect the privacy of others.  
#2) Think before you type.  
#3) With great power comes great responsibility.  
  
[sudo] password for Bellatrix:  
[Bellatrix@serveransible ~]$ sudo passwd Malfoy  
Changing password for user Malfoy.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[Bellatrix@serveransible ~]$ █  
  
[Bellatrix@serveransible ~]$ sudo useradd -m -d /home/Seamus Seamus  
[Bellatrix@serveransible ~]$ sudo passwd Seamus  
Changing password for user Seamus.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[Bellatrix@serveransible ~]$ █
```

2. Mengelompokkan Malfoy dan Seamus pada group siswa.

Kita membuat grup baru dengan nama SiswaBadGuys dan menset akun ke grup

```
[Bellatrix@serveransible ~]$ sudo groupadd SiswaBadGuys  
[Bellatrix@serveransible ~]$ sudo usermod -a -G SiswaBadGuys Malfoy  
[Bellatrix@serveransible ~]$ sudo usermod -a -G SiswaBadGuys Seamus
```

3. Tim Voldemort melakukan kecurangan, dengan memberikan wewenang kepada Malfoy dan Seamus untuk dapat menghentikan seluruh proses yang dijalankan oleh pengguna Harry dan Hermione.

Tim voldemort melakukukan kecurangan dengan memberikan akses lebih

```
##  
## Allow root to run any commands anywhere  
root    ALL=(ALL)        ALL  
Dumbledore  ALL=(ALL)      ALL  
Voldemort   ALL=(ALL)      ALL  
Snape      ALL=/sbin/adduser, /bin/passwd, /sbin/usermod, /sbin/groupadd  
Sirius     ALL=/sbin/adduser, /bin/passwd, /sbin/usermod, /sbin/groupadd  
Lucius     ALL=/sbin/adduser, /bin/passwd, /sbin/usermod, /sbin/groupadd  
Bellatrix   ALL=/sbin/adduser, /bin/passwd, /sbin/usermod, /sbin/groupadd  
Malfoy     ALL=/bin/pkill -u Harry, /bin/pkill -u Hermione  
Seamus     ALL=/bin/pkill -u Harry, /bin/pkill -u Hermione
```

Lucius memberikan perintah kepada Malfoy dan Seamus untuk melakukan operasi berikut:

1. Malfoy harus menghentikan seluruh proses yang dijalankan oleh Harry.
- Malfoy menghentikan semua proses Harry dengan mengetik perintah dibawah

```
[Malfoy@serveransible ~]$ sudo pkill -u Harry  
[sudo] password for Malfoy:  
[Malfoy@serveransible ~]$ █
```

Dan kita dapat melihat bahwa proses yang sedang dilakukan Harry di terminated

```
[Harry@serveransible ~]$ nano rio.c  
Received SIGHUP or SIGTERM  
[Harry@serveransible ~]$
```

2. Seamus harus menghentikan seluruh proses yang dijalankan oleh Hermione.

Seamus juga menterminated proses yang sedang dilakukan Hermione

```
[Seamus@serveransible ~]$ sudo pkill -u Hermione
```

We trust you have received the usual lecture from the local System Administrator. It usually boils down to these three things:

```
#1) Respect the privacy of others.  
#2) Think before you type.  
#3) With great power comes great responsibility.
```

```
[sudo] password for Seamus:  
[Seamus@serveransible ~]$
```

Proses Hermione telah di terminated

```
[Hermione@serveransible ~]$ nano 038.c  
Received SIGHUP or SIGTERM  
[Hermione@serveransible ~]$ █
```

Catatan: untuk melakukan instruksi 1 dan 2, anda dapat menjalankan program yang sudah anda buatkan pada praktikum mengenai proses untuk Harry dan Hermione.

Dumbledore mengetahui bahwa Malfoy dan Seamus melakukan operasi yang seharusnya tidak boleh mereka lakukan. Bagaimanakah Dumbledore dapat mengetahui Malfoy dan Seamus menghentikan proses Harry dan Hermione, dan Bagaimana caranya Dumbledore dapat mengetahui siapa pengguna yang memberikan wewenang tersebut kepada Malfoy dan

seamus?

Jawab: dengan cara mengetik command seperti dibawah

```
sudo cat /var/log/secure | grep 'Malfoy|Seamus'
```

Output:

```
Sep 20 20:09:52 serveransible sudo: Malfoy : TTY=pts/0 ; PWD=/home/Malfoy ; USER=root ; COMMAND=/bin/pkill -u Harry
Sep 20 20:12:40 serveransible su: pam_unix(su:session): session opened for user Malfoy by Voldemort(uid=1020)
Sep 20 20:12:46 serveransible sudo: Malfoy : TTY=pts/1 ; PWD=/home/Voldemort ; USER=root ; COMMAND=/bin/pkill -u Harry
Sep 20 20:15:18 serveransible su: pam_unix(su:session): session opened for user Seamus by Voldemort(uid=1028)
Sep 20 20:15:46 serveransible sudo: Seamus : TTY=pts/1 ; PWD=/home/Seamus ; USER=root ; COMMAND=/bin/pkill -u Hermione
Sep 20 20:17:26 serveransible su: pam_unix(su:session): session closed for user Malfoy
[Dumbledore@serveransible ~]$ █
```

Melihat kondisi kecurangan dari tim Voldemort, maka Dumbledore melakukan hal berikut:

1. Menghapus wewenang Malfoy dan Seamus sebagai superuser.

Jawab: dengan masuk ke /etc/sudoers dan menghapus wewenang Malfoy dan Seamus

```
root    ALL=(ALL)        ALL
Dumbledore   ALL=(ALL)        ALL
Voldemort   ALL=(ALL)        ALL
Snape      ALL=/sbin/adduser, /bin/passwd, /sbin/usermod, /sbin/groupadd
Sirius     ALL=/sbin/adduser, /bin/passwd, /sbin/usermod, /sbin/groupadd
Lucius     ALL=/sbin/adduser, /bin/passwd, /sbin/usermod, /sbin/groupadd
Bellatrix   ALL=/sbin/adduser, /bin/passwd, /sbin/usermod, /sbin/groupadd
#Malfoy     ALL=/bin/pkill -u Harry, /bin/pkill -u Hermione
#Seamus     ALL=/bin/pkill -u Harry, /bin/pkill -u Hermione
## Allows members of the 'sys' group to run networking, software,
...
```

2. Memerintahkan Snape dan Sirius untuk menghapus akun dari Lucius dan Bellatrix.

Bagaimanakah cara Dumbledore untuk memberikan wewenang kepada Snape dan Sirius untuk menghapus akun Lucius dan Bellatrix?

Dengan cara memberikan wewenang kepada Snape dan Sirius

```
Voldemort  ALL=(ALL)        ALL
Snape      ALL=/sbin/adduser, /bin/passwd, /sbin/usermod, /sbin/groupadd, /sbin/userdel
Sirius     ALL=/sbin/adduser, /bin/passwd, /sbin/usermod, /sbin/groupadd, /sbin/userdel
Lucius     ALL=/sbin/adduser, /bin/passwd, /sbin/usermod, /sbin/groupadd
```

Buktikan bahwa Snape dan Sirius pada akhirnya berhasil menghapus akun dari Lucius dan Bellatrix.

Snape dan Sirius berhasil menghapus Lucius dan Bellatrix

```
[Snape@serveransible ~]$ sudo userdel -r Lucius
[sudo] password for Snape:
[Snape@serveransible ~]$ su Sirius
Password:
[Sirius@serveransible Snape]$ cd
[Sirius@serveransible ~]$ sudo userdel -r Bellatrix
[sudo] password for Sirius:
[Sirius@serveransible ~]$ █
```

Lucius dan Bellatrix berhasil dihapus

```
[Sirius@serveransible ~]$ su Lucius
su: user Lucius does not exist
[Sirius@serveransible ~]$ su Bellatrix
su: user Bellatrix does not exist
[Sirius@serveransible ~]$ █
```

Mengetahui bahwa Voldemort pasti memiliki cara lainnya untuk menghapus Harry, maka

Dumbledore memberikan wewenang kepada Hermione untuk melindungi Harry, dengan cara berikut:

1. Mendaftarkan Hermione menjadi superuser.

Jawab: kita masuk ke file /etc/sudoers kembali dan memberikan wewenang kepada Hermione

2. Hermione diberikan wewenang untuk menghapus akun pengguna lainnya.

Hermione telah mendapatkan wewenang

```
#Seamus      ALL=/bin/pkill -u Harry, /bin/pkill -u Hermione
Hermione    ALL=/sbin/userdel
```

Setelah diberikan wewenang oleh Dumbledore, maka Hermione menghapus akun Malfoy dan Seamus.

Jawab: Hermione berhasil menghapus Malfoy dan Seamus

```
[Hermione@serveransible ~]$ sudo userdel -r Malfoy
[sudo] password for Hermione:
[Hermione@serveransible ~]$ sudo userdel -r Seamus
[Hermione@serveransible ~]$ █
```

Bukti telah terhapus:

```
[Hermione@serveransible ~]$ su Malfoy
su: user Malfoy does not exist
[Hermione@serveransible ~]$ su Seamus
su: user Seamus does not exist
[Hermione@serveransible ~]$ █
```

Voldemort, tetap tidak tinggal diam, dan kemudian tetap mencoba untuk menghapus akun Harry, namun dengan meminta pengguna yang lain melakukannya, dengan cara sebagai berikut:

1. Menciptakan pengguna baru yaitu Squirrel, lengkap dengan password dan home directorynya.

Voldemort membuat user baru seperti gambar dibawah

```
[Voldemort@serveransible ~]$ sudo useradd -m -d /home/Squirrel Squirrel
[sudo] password for Voldemort:
[Voldemort@serveransible ~]$ sudo passwd Squirrel
Changing password for user Squirrel.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

Serta menset user baru ke dalam grup

```
[Voldemort@serveransible ~]$ sudo usermod -aG SiswaBadGuys Squirrel
[Voldemort@serveransible ~]$ █
```

2. Mendaftarkan Squirrel menjadi superuser dan memberikan wewenang untuk menghapus pengguna lainnya.

Serta memberikan wewenang kepada Squirrel untuk menghapus user lain

```
Squirrel    ALL=/sbin/userdel
## Allows members of the 'sys' group to
```

Dumbledore, mengetahui terdapat pengguna baru yang diciptakan oleh Voldemort. Bagaimanakah Dumbledore bisa mengetahui hal ini? Buktikan!

Jawab: dengan mengetik command tail /etc/passwd

```
[Dumbledore@serveransible ~]$ tail /etc/passwd
rio38:x:1016:1021::/home/rio38:/bin/bash
user_1:x:1017:1022::/home/user_1:/bin/bash
user_2:x:1018:1023::/home/user_2:/bin/bash
Dumbledore:x:1019:1024::/home/Dumbledore:/bin/bash
Voldemort:x:1020:1025::/home/Voldemort:/bin/bash
Sirius:x:1022:1030::/home/Sirius:/bin/bash
Snape:x:1023:1031::/home/Snape:/bin/bash
Harry:x:1024:1032::/home/Harry:/bin/bash
Hermione:x:1025:1033::/home/Hermione:/bin/bash
Squirrel:x:1026:1026::/home/Squirrel:/bin/bash
[Dumbledore@serveransible ~]$
```

Kemudian, Dumbledore segera melindungi Harry dengan melakukan hal berikut:

1. Mendaftarkan Harry menjadi superuser.

Jawab: dengan masuk kembali ke /etc/sudoers dan mendaftarkan Harry

```
## Allow root to run any commands anywhere
root    ALL=(ALL)      ALL
Dumbledore  ALL=(ALL)     ALL
Voldemort   ALL=(ALL)     ALL
Harry       ALL=(ALL)     ALL
```

2. Memberikan wewenang yang setara dengan root, agar dapat menghapus Voldemort dan Squirell.

Harry sudah setara dengan root

```
## Allow root to run any commands anywhere
root    ALL=(ALL)      ALL
Dumbledore  ALL=(ALL)     ALL
Voldemort   ALL=(ALL)     ALL
Harry       ALL=(ALL)     ALL
```

Setelah diberikan wewenang oleh Dumbledore, maka Harry menghapus pengguna Voldemort dan Squirell dan beserta seluruh home directorynya.

Jawab: harry berhasil menghapus Voldemort dan Squirrel

```
[Harry@serveransible ~]$ sudo userdel -r Voldemort
[sudo] password for Harry:
[Harry@serveransible ~]$ sudo userdel -r Squirrel
[Harry@serveransible ~]$
```

Bukti telah terhapus:

```
[Harry@serveransible ~]$ su Voldemort
su: user Voldemort does not exist
[Harry@serveransible ~]$ su Squirrel
su: user Squirrel does not exist
[Harry@serveransible ~]$
```

# **BAB 4. ACCESS CONTROL, ROOTLY POWER AND FILE SYSTEM - PART 2**

## Tugas

Dengan menggunakan panduan praktikum yang telah dijelaskan pada bagian A.1 sampai dengan A.5, lakukanlah pengaturan atau konfigurasi pada server Anda sesuai dengan skenario berikut. Anda harus menyediakan laporan dengan menunjukkan desain kerja Anda serta hasil screenshot dari konfigurasi yang Anda terapkan serta konfirmasi hasil kebenaran konfigurasi.

## Skenario

Anda seorang sistem administrator dari sebuah universitas yang bernama Beyond Imagination. Tugas Anda adalah untuk mendesain dan mengkonfigurasi sebuah file server berbasis Linux untuk menentukan akses terhadap objek (file atau direktori) sesuai dengan ketentuan yang telah ditetapkan bersama oleh jajaran universitas.

Pada file server universitas akan terdapat beberapa direktori kerja untuk masing-masing unitkerja, dan tugas Anda adalah menciptakan folder sesuai dengan hierarki direktori kerja dan menetapkan hak akses direktori kerja sesuai dengan yang telah ditetapkan.

Universitas Beyond Imagination memiliki struktur organisasi yang terdiri dari Rektor sebagai kepala universitas tertinggi yang memiliki tiga wakil rektor yaitu Wakil Rektor 1 (WR1), Wakil Rektor 2 (WR2), Wakil Rektor 3 (WR3). Setiap wakil rektor memiliki unit yang dipimpin. WR1 memimpin unit Kemahasiswaan dan Keasramaan, WR2 memimpin unit SDM dan Keuangan, WR3 memimpin unit Promosi dan Kerjasama.

Untuk unit yang berada dibawah setiap WR akan mengelola kegiatan yang berbeda-beda. Pada WR1 untuk unit Kemahasiswaan akan mengelola bagian kerja terkait Pembinaan Karakter dan Kegiatan Mahasiswa, sedangkan untuk unit Keasramaan akan mengelola bagian kerja yang berkaitan dengan Pengalokasian Asrama dan Kegiatan Asrama. Pada WR2 untuk unit SDM akan mengelola bagian kerja yang berkaitan dengan Artefak Dosen dan Artefak Staf, sedangkan unit Keuangan akan mengelola bagian kerja terkait dengan RKA Unit dan Bursar Mahasiswa. Pada WR3 untuk unit Promosi akan mengelola bagian kerja mengenai Strategi Promosi, sedangkan unit Kerjasama akan mengelola bagian kerja terkait dengan Kerjasama.

Untuk melancarkan kerja, setiap unit memerlukan direktori kerja. Oleh karena itu,

Rektor akan memiliki direktori kerja sendiri, unit WR1, WR2 dan WR3 akan memiliki direktori kerja dengan masing-masing WR sebagai direktori induknya dengan subdirektori sesuai dengan nama masing-masing unit yang berada dibawah kepemimpinan masing-masing WR tersebut. Setiap unit yang berada dibawah WR akan memiliki direktori sesuai dengan bagian kerja dari masing-masing unit.

Akses terhadap untuk masing-masing direktori kerja unit perlu diatur sesuai dengan ketentuan yang ditetapkan. Sebelum hak akses ditetapkan ke setiap direktori kerja, diperlukan adanya pembentukan grup yang akan mengelompokkan seluruh anggota

unit. Nama grup harus disesuaikan dengan nama setiap unit kerja. Tabel berikut menampilkan data unit kerja dan pengguna pada setiap unit.

Unit	Anggota
Rektor	Downey
WR1	Scarlett & Johhanson
Kemahasiswaan	Ruffalo & Mark
Keasramaan	Hermsworth & Chris
WR2	Evans & Chris
SDM	Hiddleston & Tom
Keuangan	Witwicky & Sam
WR3	Renner & Jeremy
Promosi	Fox & Meghan
Kerjasama	Mackie & Anthony

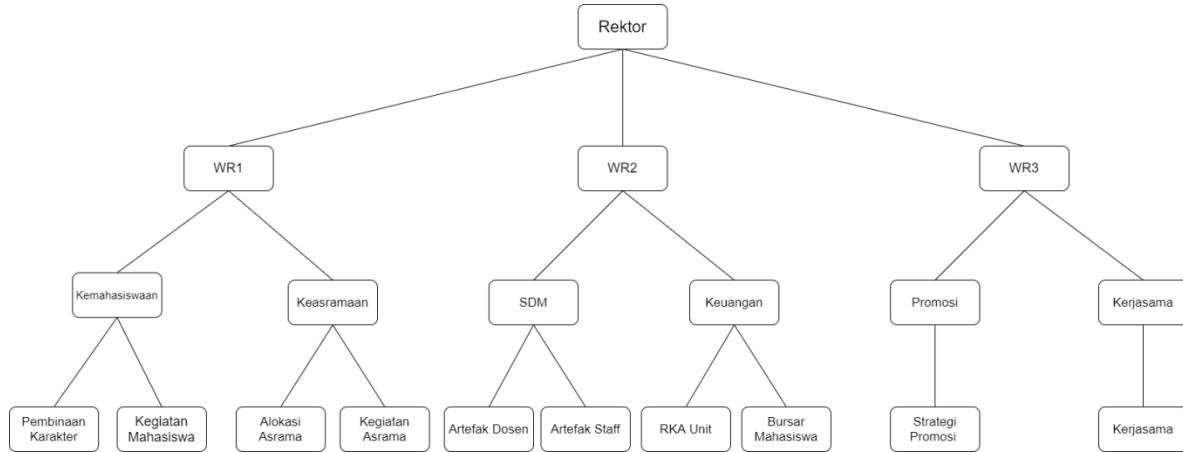
Khusus untuk anggota WR1, WR2 dan WR3 masing-masing WR memiliki sekretaris. Secara berurutan, nama sekretaris untuk setiap WR adalah Johhanson, Chris dan Jeremy.

Setelah setiap pengguna ditetapkan ke group unit masing-masing, ditetapkan mengenai kepemilikan dari setiap direktori kerja. Rektor akan sebagai pemilik mutlak direktori kerjanya, begitu juga dengan WR1, WR2 dan WR3 akan sebagai pemilik mutlak direktori kerjanya beserta dengan subdirektori didalamnya. Pada setiap direktori WR masing-masing WR dan sekretaris yang terkait akan memiliki direktori kerja pribadi. Rektor dan seluruh WR beserta sekretarisnya akan memiliki hak akses penuh ke direktori kerja masing-masing.

Selanjutnya, direktori kerja akan dapat diakses secara penuh oleh setiap grup unit yang berada dibawah masing-masing WR. Anggota unit yang berada dari luar unit yang dipimpin oleh masing-masing WR, tidak diperbolehkan mengakses direktori kerja. Kemudian, untuk masing-masing direktori kerja yang berada dibawah pengelolaan masing-masing WR, ditetapkan bahwa hanya grup unit yang bersangkutan saja yang dapat mengakses direktori kerja tersebut. Unit lain,

meskipun berada dibawah kepemimpinan WR yang sama, tidak boleh saling mengakses direktori kerja unit tersebut.

## Design Struktural



Namun, terdapat beberapa ketentuan khusus yang perlu diperhatikan, sebagai berikut:

1. Direktori kerja Rektor tidak dapat diakses oleh siapapun kecuali oleh Rektorsendiri.
2. WR mendapatkan hak akses penuh untuk folder sekretaris, tetapi sekretaris tidak dapat mengakses direktori kerja pribadi WR.
3. Dikarenakan, bagian kerja kemahasiswaan sering berkaitan dengan bagiankeasramaan, maka khusus kepada ketua pembina asrama diberikan akses untuk membaca file (buatkan contoh file program\_bina\_karakter.txt) yang berada pada bagian kerja pembinaan karakter.
4. Begitu juga dengan bagian kerja RKA Unit harus dapat diakses oleh seluruh unit dengan hak akses penuh. Selanjutnya, dikarenakan WR1 juga memerlukan informasi strategi promosi dan juga kerjasama yang dimiliki oleh universitas, maka khusus untuk WR1 (tidak termasuk sekretaris) diberikan hak akses untuk membaca file yang berada pada direktori kerja bagian promosi dan kerjasama.
5. Kondisi berikutnya adalah, khusus untuk bagian kerja SDM sekretaris dari WR2 tidak diberikan hak akses sama sekali.
6. Khusus untuk unit keuangan bagian kerja Bursar Mahasiswa, Sam dan sekretaris WR2 tidak mendapatkan hak akses sama sekali pada direktori tersebut.
7. Dikarenakan masing-masing WR harus membutuhkan akses cepat untuk beberapa informasi tertentu maka dibutuhkan hard link atau soft link yang ada pada direktori kerja pribadi merujuk ke beberapa file berikut. Untuk WR1 dibutuhkan link ke file program\_bina\_karakter.txt pada direktori kerja pembinaan karakter, WR2 dibutuhkan link ke file RKA\_unit\_kemahasiswaan.txt pada direktori kerja RKA Unit, dan pada WR3 dibutuhkan link ke file kerjasama\_dalam\_dan\_luar\_negri.txt yang berada pada direktori kerja Kerjasama.

Jawab:

1. Membuat direktori kampus

```
[root@serveransible ~]# mkdir Universitas_Beyond_Imagination  
[root@serveransible ~]#
```

2. Kemudian masuk kedalam direktori dan membuat direktori Rektor, WR1, WR2, WR3.

```
[root@serveransible Universitas_Beyond_Imagination]# mkdir Rektor | mkdir WR1 | mkdir WR2  
[root@serveransible Universitas_Beyond_Imagination]#
```

3. Kemudian di dalam direktori WR1 kita membuat direktori Kemahasiswaan dan Keasramaan

```
[root@serveransible WR1]# mkdir Kemahasiswaan | mkdir Keasramaan  
[root@serveransible WR1]#
```

4. Kemudian kita keluar dari direktori WR1 dan masuk ke direktori WR2 dan membuat direktori SDM dan Keuangan.

```
[root@serveransible Universitas_Beyond_Imagination]# cd WR2  
[root@serveransible WR2]# mkdir SDM | mkdir Keuangan  
[root@serveransible WR2]#
```

5. Kemudian kita keluar dari WR2 dan masuk ke WR3 dan membuat direktori Promosi dan Kerjasama

```
[root@serveransible Universitas_Beyond_Imagination]# mkdir WR3  
[root@serveransible Universitas_Beyond_Imagination]# cd WR3  
[root@serveransible WR3]# mkdir Promosi | mkdir Kerjasama  
[root@serveransible WR3]#
```

6. Kemudian kita kembali ke direktori WR1 dan masuk ke direktori Kemahasiswaan dan Keasramaan dan membuat direktori Bina\_Karakter dan Kegiatan\_mahasiswa dalam direktori Kemahasiswaan dan direktori Alokasi\_asrama dan Kegiatan\_asrama di dalam Keasramaan

```
[root@serveransible Universitas_Beyond_Imagination]# cd WR1  
[root@serveransible WR1]# cd Kemahasiswaan  
[root@serveransible Kemahasiswaan]# mkdir Bina_karakter | mkdir Kegiatan_mahasiswa  
[root@serveransible Kemahasiswaan]# cd ..  
[root@serveransible WR1]# cd keasramaan  
bash: cd: keasramaan: No such file or directory  
[root@serveransible WR1]# cd Keasramaan  
[root@serveransible Keasramaan]# mkdir Alokasi_asrama | mkdir Kegiatan_asrama  
[root@serveransible Keasramaan]#
```

7. Kemudian kita masuk ke direktori WR2 dan masuk ke direktori SDM dan Keuangan. didalam direktori SDM kita membuat 2 direktori baru yaitu artefak\_dosen dan artefak\_staff. di dalam direktori Keuangan kita membuat 2 direktori baru yaitu RKA\_unit dan Bursar\_mahasiswa.

```
[root@serveransible WR1]# cd ..  
[root@serveransible Universitas_Beyond_Imagination]# cd WR2  
[root@serveransible WR2]# cd SDM  
[root@serveransible SDM]# mkdir artefak_dosen | mkdir artefak_staff  
[root@serveransible SDM]# cd ..  
[root@serveransible WR2]# cd Keuangan  
[root@serveransible Keuangan]# mkdir RKA_unit | mkdir Bursar_mahasiswa
```

8. Kemudian kita masuk ke WR3 dan masuk ke direktori Promosi dan Kerjasama. didalam direktori Promosi kita membuat direktori Strategi\_promosi dan didalam direktori kerjasama kita membuat direktori kerjasama

```
[root@serveransible ~]# cd ..  
[root@serveransible Universitas_Beyond_Imagination]# cd WR3  
[root@serveransible WR3]# cd Promosi  
[root@serveransible Promosi]# mkdir Strategi_promosi  
[root@serveransible Promosi]# cd ..  
bash: cd..: command not found...  
[root@serveransible Promosi]# cd ..  
[root@serveransible WR3]# cd Kerjasama  
[root@serveransible Kerjasama]# mkdir Kerjasama  
[root@serveransible Kerjasama]# █
```

9. Sekarang kita dapat membuat grup yang sesuai dengan nama unit

```
[root@serveransible ~]# cd  
[root@serveransible ~]# groupadd Rektor | groupadd WR1 | groupadd WR2 | groupadd WR3  
[root@serveransible ~]# groupadd Kemahasiswaan | groupadd Keasramaan  
[root@serveransible ~]# groupadd SDM | groupadd Keuangan  
[root@serveransible ~]# groupadd Promosi | groupadd Kerjasama  
[root@serveransible ~]# █
```

10. Kemudian kita membuat user sesuai dengan ketentuan.

```
[root@serveransible ~]# useradd -m -d /home/Downey Downey  
[root@serveransible ~]# passwd Downey  
Changing password for user Downey.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@serveransible ~]# useradd -m -d /home/Scarlett Scarlett  
[root@serveransible ~]# passwd Scarlett  
Changing password for user Scarlett.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@serveransible ~]# useradd -m -d /home/Johhanson Johhanson  
[root@serveransible ~]# passwd Johhanson  
Changing password for user Johhanson.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@serveransible ~]# useradd -m -d /home/Ruffalo Ruffalo  
[root@serveransible ~]# passwd Ruffalo  
Changing password for user Ruffalo.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@serveransible ~]# useradd -m -d /home/Mark Mark  
[root@serveransible ~]# passwd Mark  
Changing password for user Mark.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@serveransible ~]#
```

```
[root@serveransible ~]# useradd -m -d /home/Hermsworth Hermsworth
[root@serveransible ~]# passwd Hermsworth
Changing password for user Hermsworth.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Chris Chris
[root@serveransible ~]# passwd Chris
Changing password for user Chris.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Evans Evans
[root@serveransible ~]# passwd Evans
Changing password for user Evans.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Hiddleston Hiddleston
[root@serveransible ~]# passwd Hiddleston
Changing password for user Hiddleston.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Tom Tom
[root@serveransible ~]# passwd Tom
Changing password for user Tom.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]#
```

```
[root@serveransible ~]# useradd -m -d /home/Witwickly Witwickly
[root@serveransible ~]# passwd Witwickly
Changing password for user Witwickly.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Sam Sam
[root@serveransible ~]# passwd Sam
Changing password for user Sam.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Renner Renner
[root@serveransible ~]# passwd Renner
Changing password for user Renner.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Jeremy Jeremy
[root@serveransible ~]# passwd Jeremy
Changing password for user Jeremy.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Fox Fox
[root@serveransible ~]# passwd Fox
Changing password for user Fox.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]#
```

```
[root@serveransible ~]# useradd -m -d /home/Meghan Meghan
[root@serveransible ~]# passwd Meghan
Changing password for user Meghan.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Mackie Mackie
[root@serveransible ~]# passwd Mackie
Changing password for user Mackie.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Anthony Anthony
[root@serveransible ~]# passwd Anthony
Changing password for user Anthony.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]#
```

11. Kemudian kita memasukkan user yang telah dibuat kedalam grup sesuai dengan ketentuan

```
[root@serveransible ~]# usermod -aG Rektor Downey
[root@serveransible ~]# usermod -aG WR1 Scarlett | usermod -aG WR1 Johhanson
[root@serveransible ~]# usermod -aG Kemahasiswaan Ruffalo | usermod -aG Kemahasiswaan Mark
[root@serveransible ~]# usermod -aG Keasramaan Ruffalo | usermod -aG Keasramaan Chris
[root@serveransible ~]# usermod -aG WR2 Evans | usermod -aG WR2 Chris
[root@serveransible ~]# usermod -aG SDM Hiddleston | usermod -aG SDM Tom
[root@serveransible ~]# usermod -aG Keuangan Witwicky | usermod -aG Keuangan Sam
[root@serveransible ~]# usermod -aG WR3 Renner | usermod -aG WR3 Jeremy
[root@serveransible ~]# usermod -aG Promosi Fox | usermod -aG Promosi Meghan
[root@serveransible ~]# usermod -aG Kerjasama Mackie | usermod -aG Kerjasama Anthony
[root@serveransible ~]#
```

12. Direktori kerja Rektor tidak dapat diakses oleh siapapun kecuali oleh Rektorsendiri.

```
[root@serveransible Universitas_Beyond_Imagination]# chown Downey Rektor
[root@serveransible Universitas_Beyond_Imagination]# chmod 700 Rektor
[root@serveransible Universitas_Beyond_Imagination]#
```

Penjelasan: kita menset Downey selaku rektor sebagai owner direktori Rektor dan memberikan akses penuh, tetapi grup dan other tidak dapat mengaksesnya

13. WR mendapatkan hak akses penuh untuk folder sekretaris, tetapi sekretaris tidak dapat mengakses direktori kerja pribadi WR.

```
[root@serveransible Universitas_Beyond_Imagination]# chown Scarlett:WR1 WR1
[root@serveransible Universitas_Beyond_Imagination]# chmod 700 WR1
[root@serveransible Universitas_Beyond_Imagination]# chown Evans:WR2 WR2
[root@serveransible Universitas_Beyond_Imagination]# chmod 700 WR2
[root@serveransible Universitas_Beyond_Imagination]# chown Renner:WR3 WR3
[root@serveransible Universitas_Beyond_Imagination]# chmod 700 WR3
[root@serveransible Universitas_Beyond_Imagination]#
```

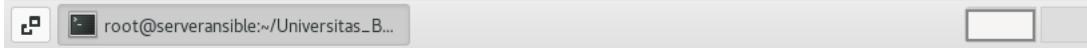
Penjelasan: kita menset Scarlett, Evans, Renner sebagai owner dari direktori WR1, WR2, WR3 namun sekretaris tidak dapat mengakses.

```
[root@serveransible Universitas_Beyond_Imagination]# mkdir Sekretaris_WR1 | mkdir Sekretaris_WR2 | mkdir Sekretaris_WR3
[root@serveransible Universitas_Beyond_Imagination]# 
[root@serveransible Universitas_Beyond_Imagination]# chown Johhanson:WR1 Sekretaris_WR1
[root@serveransible Universitas_Beyond_Imagination]# chmod 770 Sekretaris_WR1
[root@serveransible Universitas_Beyond_Imagination]# chown Chris:WR2 Sekretaris_WR2
[root@serveransible Universitas_Beyond_Imagination]# chmod 770 Sekretaris_WR2
[root@serveransible Universitas_Beyond_Imagination]# chown Jeremy:WR3 Sekretaris_WR3
[root@serveransible Universitas_Beyond_Imagination]# chmod 770 Sekretaris_WR3
[root@serveransible Universitas_Beyond_Imagination]#
```

Penjelasan: Johhanson, Chris, dan Jeremy merupakan sekretaris dari WR sehingga mereka menjadi owner dari direktori mereka dan menset grup dapat mengakses direktori yang sama.

14. Dikarenakan, bagian kerja kemahasiswaan sering berkaitan dengan bagian keasramaan, maka khusus kepada ketua pembina asrama diberikan akses untuk membaca file (buatkan contoh file program\_bina\_karakter.txt) yang berada pada bagian kerja pembinaan karakter.

```
[root@serveransible Bina_karakter]# setfacl -m g:Keasramaan:r program_bina_karakter.txt
[root@serveransible Bina_karakter]#
```



15. Begitu juga dengan bagian kerja RKA Unit harus dapat diakses oleh seluruh unit dengan hak akses penuh. Selanjutnya, dikarenakan WR1 juga memerlukan informasi strategi promosi dan juga kerjasama yang dimiliki oleh universitas, maka khusus untuk WR1 (tidak termasuk sekretaris) diberikan hak akses untuk membaca file yang berada pada direktori kerja bagian promosi dan kerjasama.

```
[root@serveransible Keuangan]# chmod 770 RKA_unit
[root@serveransible Keuangan]#
```

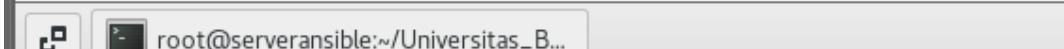
Penjelasan: RKA unit dapat diakses oleh semua unit.

```
[root@serveransible Universitas_Beyond_Imagination]# setfacl -m u:Scarlett:r WR3
[root@serveransible Universitas_Beyond_Imagination]#
```

Penjelasan: WR1 dapat membaca file didirektori WR3

16. Kondisi berikutnya adalah, khusus untuk bagian kerja SDM sekretaris dari WR2 tidak diberikan hak akses sama sekali.

```
[root@serveransible WR2]# setfacl -m g:WR1:rwx SDM
[root@serveransible WR2]# setfacl -m g:WR3:rwx SDM
[root@serveransible WR2]# setfacl -m u:Evans:rwx SDM
[root@serveransible WR2]#
```



17. Khusus untuk unit keuangan bagian kerja Bursar Mahasiswa, Sam dan sekretaris WR2 tidak mendapatkan hak akses sama sekali pada direktori tersebut.

```
[root@serveransible WR2]# chown Witwicky Keuangan
[root@serveransible WR2]# setfacl -m u:Scarlett:rwx Keuangan
[root@serveransible WR2]# setfacl -m g:WR3:rwx Keuangan
[root@serveransible WR2]# chmod 770 Keuangan
[root@serveransible WR2]#
```

Penjelasan: semua unit dapat mengakses kecuali sam dan Johhanson

18. Dikarenakan masing-masing WR harus membutuhkan akses cepat untuk beberapa informasi tertentu maka dibutuhkan hard link atau soft link yang ada pada direktori kerja pribadi merujuk ke beberapa file berikut. Untuk WR1 dibutuhkan link ke file program\_bina\_karakter.txt pada direktori kerja pembinaan karakter, WR2 dibutuhkan link ke file RKA\_unit\_kemahasiswaan.txt pada direktori kerja RKA Unit, dan pada WR3 dibutuhkan link ke file kerjasama\_dalam\_dan\_luar\_negri.txt yang berada pada direktori kerja Kerjasama.

```
[root@serveransibile Kemahasiswaan]# cd Bina_karakter
[root@serveransibile Bina_karakter]# nano program_bina_karakter.txt
[root@serveransibile Bina_karakter]#
```

Penjelasan: pembuatan program\_bina\_karakter.txt dalam direktori Bina\_karakter.

```
[root@serveransibile WR2]# cd Keuangan
[root@serveransibile Keuangan]# cd RKA_unit
[root@serveransibile RKA_unit]# nano RKA_unit_kemahasiswaan.txt
[root@serveransibile RKA_unit]#
```

Penjelasan: pembuatan program RKA\_unit\_kemahasiswaan.txt

```
[root@serveransibile ~]# cd Universitas_Beyond_Imagination
[root@serveransibile Universitas_Beyond_Imagination]# cd WR3
[root@serveransibile WR3]# cd Kerjasama
[root@serveransibile Kerjasama]# cd Kerjasama
[root@serveransibile Kerjasama]# ls
[root@serveransibile Kerjasama]# nano kerjasama_dalam_dan_luar_negeri.txt
[root@serveransibile Kerjasama]#
```

Penjelasan: proses pembuatan file kerjasama\_dalam\_dan\_luar\_negeri.txt pada direktori kerjasama.

```
[root@serveransibile Universitas_Beyond_Imagination]# ln -s /Universitas_Beyond_Imagination/WR1/Kemahasiswaan/Bina_karakter/program_bina_karakter.txt /WR1
[root@serveransibile Universitas_Beyond_Imagination]#
```

```
{_Imagination]# ln -s /Universitas_Beyond_Imagination/WR1/Kemahasiswaan/Bina_karakter/program_bina_karakter.txt /WR1
{_Imagination]#
```

Penjelasan: command diatas merupakan soft link ke file program\_bina\_karakter.txt

```
[root@serveransibile ~]# ln -s /Universitas_Beyond_Imagination/WR2/Keuangan/RKA_unit/RKA_unit_kemahasiswaan.txt /WR2
[root@serveransibile ~]# ls -l /Universitas_Beyond_Imagination/WR2/Keuangan/RKA_unit/RKA_unit_kemahasiswaan.txt /WR2
ls: cannot access /Universitas_Beyond_Imagination/WR2/Keuangan/RKA_unit/RKA_unit_kemahasiswaan.txt: No such file or directory
lrwxrwxrwx 1 root root 80 Sep 23 14:12 /WR2 -> /Universitas_Beyond_Imagination/WR2/Keuangan/RKA_unit/RKA_unit_kemahasiswaan.txt
[root@serveransibile ~]#
```

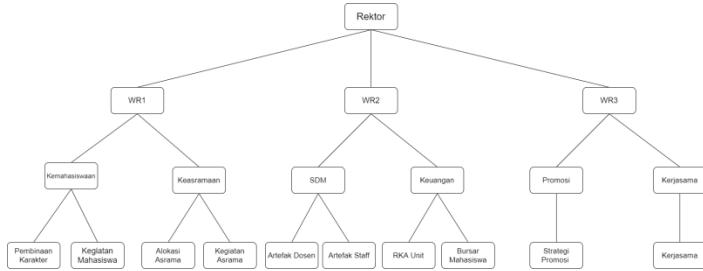
Penjelasan: command diatas merupakan soft link ke file program RKA\_unit\_kemahasiswaan.txt

```
[root@serveransibile ~]# ls -l /Universitas_Beyond_Imagination/WR3/Kerjasama/Kerjasama_dalam_dan_luar_negeri.txt /WR3
ls: cannot access /Universitas_Beyond_Imagination/WR3/Kerjasama/Kerjasama_dalam_dan_luar_negeri.txt: No such file or directory
lrwxrwxrwx 1 root root 91 Sep 23 14:25 /WR3 -> /Universitas_Beyond_Imagination/WR3/Kerjasama/Kerjasama_dalam_dan_luar_negeri.txt
[root@serveransibile ~]#
```

Penjelasan: command diatas merupakan soft link ke file kerjasama\_dalam\_dan\_luar\_negeri.txt

# BAB 5. FILESYSTEM AND LOGICAL VOLUME MANAGER

Design Struktural



Namun, terdapat beberapa ketentuan khusus yang perlu diperhatikan, sebagai berikut:

1. Direktori kerja Rektor tidak dapat diakses oleh siapapun kecuali oleh Rektorsendiri.
2. WR mendapatkan hak akses penuh untuk folder sekretaris, tetapi sekretaris tidak dapat mengakses direktori kerja pribadi WR.
3. Dikarenakan, bagian kerja kemahasiswaan sering berkaitan dengan bagian keasramaan, maka khusus kepada ketua pembina asrama diberikan akses untuk membaca file (buatkan contoh file program\_bina\_karakter.txt) yang berada pada bagian kerja pembinaan karakter.
4. Begitu juga dengan bagian kerja RKA Unit harus dapat diakses oleh seluruh unit dengan hak akses penuh. Selanjutnya, dikarenakan WR1 juga memerlukan informasi strategi promosi dan juga kerjasama yang dimiliki oleh universitas, maka khusus untuk WR1 (tidak termasuk sekretaris) diberikan hak akses untuk membaca file yang berada pada direktori kerja bagian promosi dan kerjasama.
5. Kondisi berikutnya adalah, khusus untuk bagian kerja SDM sekretaris dari WR2 tidak diberikan hak akses sama sekali.
6. Khusus untuk unit keuangan bagian kerja Bursar Mahasiswa, Sam dan sekretaris WR2 tidak mendapatkan hak akses sama sekali pada direktori tersebut.
7. Dikarenakan masing-masing WR harus membutuhkan akses cepat untuk beberapa informasi tertentu maka dibutuhkan hard link atau soft link yang ada pada direktori kerja pribadi merujuk ke beberapa file berikut. Untuk WR1 dibutuhkan link ke file program\_bina\_karakter.txt pada direktori kerja pembinaan karakter, WR2 dibutuhkan link ke file RKA\_unit\_kemahasiswaan.txt pada direktori kerja RKA Unit, dan pada WR3 dibutuhkan link ke file kerjasama\_dalam\_dan\_luar\_negri.txt yang berada pada direktori kerja Kerjasama.

Jawab:

1. Membuat direktori kampus

```
[root@serveransible ~]# mkdir Universitas_Beyond_Imagination  
[root@serveransible ~]#
```

2. Kemudian masuk kedalam direktori dan membuat direktori Rektor, WR1, WR2, WR3.

```
[root@serveransible Universitas_Beyond_Imagination]# mkdir Rektor | mkdir WR1 | mkdir WR2  
[root@serveransible Universitas_Beyond_Imagination]#
```

3. Kemudian di dalam direktori WR1 kita membuat direktori Kemahasiswaan dan Keasramaan

```
[root@serveransible WR1]# mkdir Kemahasiswaan | mkdir Keasramaan  
[root@serveransible WR1]#
```

4. Kemudian kita keluar dari direktori WR1 dan masuk ke direktori WR2 dan membuat direktori SDM dan Keuangan.

```
[root@serveransible Universitas_Beyond_Imagination]# cd WR2  
[root@serveransible WR2]# mkdir SDM | mkdir Keuangan  
[root@serveransible WR2]#
```

5. Kemudian kita keluar dari WR2 dan masuk ke WR3 dan membuat direktori Promosi dan Kerjasama

```
[root@serveransible Universitas_Beyond_Imagination]# mkdir WR3  
[root@serveransible Universitas_Beyond_Imagination]# cd WR3  
[root@serveransible WR3]# mkdir Promosi | mkdir Kerjasama  
[root@serveransible WR3]#
```

6. Kemudian kita kembali ke direktori WR1 dan masuk ke direktori Kemahasiswaan dan Keasramaan dan membuat direktori Bina\_Karakter dan Kegiatan\_mahasiswa dalam direktori Kemahasiswaan dan direktori Alokasi\_asrama dan Kegiatan\_asrama di dalam Keasramaan

```
[root@serveransible Universitas_Beyond_Imagination]# cd WR1  
[root@serveransible WR1]# cd Kemahasiswaan  
[root@serveransible Kemahasiswaan]# mkdir Bina_karakter | mkdir Kegiatan_mahasiswa  
[root@serveransible Kemahasiswaan]# cd ..  
[root@serveransible WR1]# cd keasramaan  
bash: cd: keasramaan: No such file or directory  
[root@serveransible WR1]# cd Keasramaan  
[root@serveransible Keasramaan]# mkdir Alokasi_asrama | mkdir Kegiatan_asrama  
[root@serveransible Keasramaan]#
```

7. Kemudian kita masuk ke direktori WR2 dan masuk ke direktori SDM dan Keuangan. didalam direktori SDM kita membuat 2 direktori baru yaitu artefak\_dosen dan artefak\_staff. di dalam direktori Keuangan kita membuat 2 direktori baru yaitu RKA\_unit dan Bursar\_mahasiswa.

```
[root@serveransible WR1]# cd ..  
[root@serveransible Universitas_Beyond_Imagination]# cd WR2  
[root@serveransible WR2]# cd SDM  
[root@serveransible SDM]# mkdir artefak_dosen | mkdir artefak_staff  
[root@serveransible SDM]# cd ..  
[root@serveransible WR2]# cd Keuangan  
[root@serveransible Keuangan]# mkdir RKA_unit | mkdir Bursar_mahasiswa
```

8. Kemudian kita masuk ke WR3 dan masuk ke direktori Promosi dan Kerjasama. didalam direktori Promosi kita membuat direktori Strategi\_promosi dan didalam direktori kerjasama kita membuat direktori kerjasama

```
[root@serveransible WR2]# cd ..  
[root@serveransible Universitas_Beyond_Imagination]# cd WR3  
[root@serveransible WR3]# cd Promosi  
[root@serveransible Promosi]# mkdir Strategi_promosi  
[root@serveransible Promosi]# cd ..  
bash: cd..: command not found...  
[root@serveransible Promosi]# cd ..  
[root@serveransible WR3]# cd Kerjasama  
[root@serveransible Kerjasama]# mkdir Kerjasama  
[root@serveransible Kerjasama]# █
```

9. Sekarang kita dapat membuat grup yang sesuai dengan nama unit

```
[root@serveransible ~]# cd  
[root@serveransible ~]# groupadd Rektor | groupadd WR1 | groupadd WR2 | groupadd WR3  
[root@serveransible ~]# groupadd Kemahasiswaan | groupadd Keasramaan  
[root@serveransible ~]# groupadd SDM | groupadd Keuangan  
[root@serveransible ~]# groupadd Promosi | groupadd Kerjasama  
[root@serveransible ~]# █
```

10. Kemudian kita membuat user sesuai dengan ketentuan.

```
[root@serveransible ~]# useradd -m -d /home/Downey Downey  
[root@serveransible ~]# passwd Downey  
Changing password for user Downey.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@serveransible ~]# useradd -m -d /home/Scarlett Scarlett  
[root@serveransible ~]# passwd Scarlett  
Changing password for user Scarlett.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@serveransible ~]# useradd -m -d /home/Johhanson Johhanson  
[root@serveransible ~]# passwd Johhanson  
Changing password for user Johhanson.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@serveransible ~]# useradd -m -d /home/Ruffalo Ruffalo  
[root@serveransible ~]# passwd Ruffalo  
Changing password for user Ruffalo.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@serveransible ~]# useradd -m -d /home/Mark Mark  
[root@serveransible ~]# passwd Mark  
Changing password for user Mark.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[root@serveransible ~]#
```

```
[root@serveransible ~]# useradd -m -d /home/Hermsworth Hermsworth
[root@serveransible ~]# passwd Hermsworth
Changing password for user Hermsworth.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Chris Chris
[root@serveransible ~]# passwd Chris
Changing password for user Chris.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Evans Evans
[root@serveransible ~]# passwd Evans
Changing password for user Evans.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Hiddleston Hiddleston
[root@serveransible ~]# passwd Hiddleston
Changing password for user Hiddleston.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Tom Tom
[root@serveransible ~]# passwd Tom
Changing password for user Tom.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]#
```

```
[root@serveransible ~]# useradd -m -d /home/Witwicky Witwicky
[root@serveransible ~]# passwd Witwicky
Changing password for user Witwicky.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Sam Sam
[root@serveransible ~]# passwd Sam
Changing password for user Sam.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Renner Renner
[root@serveransible ~]# passwd Renner
Changing password for user Renner.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Jeremy Jeremy
[root@serveransible ~]# passwd Jeremy
Changing password for user Jeremy.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Fox Fox
[root@serveransible ~]# passwd Fox
Changing password for user Fox.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]#
```

```
[root@serveransible ~]# useradd -m -d /home/Meghan Meghan
[root@serveransible ~]# passwd Meghan
Changing password for user Meghan.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Mackie Mackie
[root@serveransible ~]# passwd Mackie
Changing password for user Mackie.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]# useradd -m -d /home/Anthony Anthony
[root@serveransible ~]# passwd Anthony
Changing password for user Anthony.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
[root@serveransible ~]#
```

11. Kemudian kita memasukkan user yang telah dibuat kedalam grup sesuai dengan ketentuan

```
[root@serveransible ~]# usermod -aG Rektor Downey
[root@serveransible ~]# usermod -aG WR1 Scarlett | usermod -aG WR1 Johhanson
[root@serveransible ~]# usermod -aG Kemahasiswaan Ruffalo | usermod -aG Kemahasiswaan Mark
[root@serveransible ~]# usermod -aG Keasramaan Ruffalo | usermod -aG Keasramaan Chris
[root@serveransible ~]# usermod -aG WR2 Evans | usermod -aG WR2 Chris
[root@serveransible ~]# usermod -aG SDM Hiddleston | usermod -aG SDM Tom
[root@serveransible ~]# usermod -aG Keuangan Witwicky | usermod -aG Keuangan Sam
[root@serveransible ~]# usermod -aG WR3 Renner | usermod -aG WR3 Jeremy
[root@serveransible ~]# usermod -aG Promosi Fox | usermod -aG Promosi Meghan
[root@serveransible ~]# usermod -aG Kerjasama Mackie | usermod -aG Kerjasama Anthony
[root@serveransible ~]#
```

12. Direktori kerja Rektor tidak dapat diakses oleh siapapun kecuali oleh Rektorsendiri.

```
[root@serveransible Universitas_Beyond_Imagination]# chown Downey Rektor
[root@serveransible Universitas_Beyond_Imagination]# chmod 700 Rektor
[root@serveransible Universitas_Beyond_Imagination]#
```

Penjelasan: kita menset Downey selaku rektor sebagai owner direktori Rektor dan memberikan akses penuh, tetapi grup dan other tidak dapat mengaksesnya

13. WR mendapatkan hak akses penuh untuk folder sekretaris, tetapi sekretaris tidak dapat mengakses direktori kerja pribadi WR.

```
[root@serveransible Universitas_Beyond_Imagination]# chown Scarlett:WR1 WR1
[root@serveransible Universitas_Beyond_Imagination]# chmod 700 WR1
[root@serveransible Universitas_Beyond_Imagination]# chown Evans:WR2 WR2
[root@serveransible Universitas_Beyond_Imagination]# chmod 700 WR2
[root@serveransible Universitas_Beyond_Imagination]# chown Renner:WR3 WR3
[root@serveransible Universitas_Beyond_Imagination]# chmod 700 WR3
[root@serveransible Universitas_Beyond_Imagination]#
```

Penjelasan: kita menset Scarlett, Evans, Renner sebagai owner dari direktori WR1, WR2, WR3 namun sekretaris tidak dapat mengakses.

```
[root@serveransible Universitas_Beyond_Imagination]# mkdir Sekretaris_WR1 | mkdir Sekretaris_WR2 | mkdir Sekretaris_WR3
[root@serveransible Universitas_Beyond_Imagination]#
[root@serveransible Universitas_Beyond_Imagination]# chown Johhanson:WR1 Sekretaris_WR1
[root@serveransible Universitas_Beyond_Imagination]# chmod 770 Sekretaris_WR1
[root@serveransible Universitas_Beyond_Imagination]# chown Chris:WR2 Sekretaris_WR2
[root@serveransible Universitas_Beyond_Imagination]# chmod 770 Sekretaris_WR2
[root@serveransible Universitas_Beyond_Imagination]# chown Jeremy:WR3 Sekretaris_WR3
[root@serveransible Universitas_Beyond_Imagination]# chmod 770 Sekretaris_WR3
[root@serveransible Universitas_Beyond_Imagination]#
```

Penjelasan: Johhanson, Chris, dan Jeremy merupakan sekretaris dari WR sehingga mereka menjadi owner dari direktori mereka dan menset grup dapat mengakses direktori yang sama.

14. Dikarenakan, bagian kerja kemahasiswaan sering berkaitan dengan bagian keasramaan, maka khusus kepada ketua pembina asrama diberikan akses untuk membaca file (buatkan contoh file program\_bina\_karakter.txt) yang berada pada bagian kerja pembinaan karakter.

```
[root@serveransible Bina_karakter]# setfacl -m g:Keasramaan:r program_bina_karakter.txt  
[root@serveransible Bina_karakter]#
```

15. Begitu juga dengan bagian kerja RKA Unit harus dapat diakses oleh seluruh unit dengan hak akses penuh. Selanjutnya, dikarenakan WR1 juga memerlukan informasi strategi promosi dan juga kerjasama yang dimiliki oleh universitas, maka khusus untuk WR1 (tidak termasuk sekretaris) diberikan hak akses untuk membaca file yang berada pada direktori kerja bagian promosi dan kerjasama.

```
[root@serveransible Keuangan]# chmod 770 RKA_unit  
[root@serveransible Keuangan]#
```

Penjelasan: RKA unit dapat diakses oleh semua unit.

```
[root@serveransible Universitas_Beyond_Imagination]# setfacl -m u:Scarlett:r WR3  
[root@serveransible Universitas_Beyond_Imagination]#
```

Penjelasan: WR1 dapat membaca file didirektori WR3

16. Kondisi berikutnya adalah, khusus untuk bagian kerja SDM sekretaris dari WR2 tidak diberikan hak akses sama sekali.

```
[root@serveransible WR2]# setfacl -m g:WR1:rwx SDM  
[root@serveransible WR2]# setfacl -m g:WR3:rwx SDM  
[root@serveransible WR2]# setfacl -m u:Evans:rwx SDM  
[root@serveransible WR2]#
```

17. Khusus untuk unit keuangan bagian kerja Bursar Mahasiswa, Sam dan sekretaris WR2 tidak mendapatkan hak akses sama sekali pada direktori tersebut.

```
[root@serveransible WR2]# chown Witwicky Keuangan  
[root@serveransible WR2]# setfacl -m u:Scarlett:rwx Keuangan  
[root@serveransible WR2]# setfacl -m g:WR3:rwx Keuangan  
[root@serveransible WR2]# chmod 770 Keuangan  
[root@serveransible WR2]#
```

Penjelasan: semua unit dapat mengakses kecuali sam dan Johhanson

18. Dikarenakan masing-masing WR harus membutuhkan akses cepat untuk beberapa informasi tertentu maka dibutuhkan hard link atau soft link yang ada pada direktori kerja pribadi merujuk ke beberapa file berikut. Untuk WR1 dibutuhkan link ke file program\_bina\_karakter.txt pada direktori kerja pembinaan karakter, WR2 dibutuhkan link ke file RKA\_unit\_kemahasiswaan.txt pada direktori kerja RKA Unit, dan pada WR3 dibutuhkan link ke file kerjasama\_dalam\_dan\_luar\_negri.txt yang berada pada direktori kerja Kerjasama.

```
[root@serveransible Kemahasiswaan]# cd Bina_karakter
[root@serveransible Bina_karakter]# nano program_bina_karakter.txt
[root@serveransible Bina_karakter]#
```

Penjelasan: pembuatan program\_bina\_karakter.txt dalam direktori Bina\_karakter.

```
[root@serveransible WR2]# cd Keuangan
[root@serveransible Keuangan]# cd RKA_unit
[root@serveransible RKA_unit]# nano RKA_unit_kemahasiswaan.txt
[root@serveransible RKA_unit]#
```

Penjelasan: pembuatan program RKA\_unit\_kemahasiswaan.txt

```
[root@serveransible ~]# cd Universitas_Beyond_Imagination
[root@serveransible Universitas_Beyond_Imagination]# cd WR3
[root@serveransible WR3]# cd Kerjasama
[root@serveransible Kerjasama]# cd Kerjasama
[root@serveransible Kerjasama]# ls
[root@serveransible Kerjasama]# nano kerjasama_dalam_dan_luar_negeri.txt
[root@serveransible Kerjasama]#
```

Penjelasan: proses pembuatan file kerjasama\_dalam\_dan\_luar\_negeri.txt pada direktori kerjasama.

```
[root@serveransible Universitas_Beyond_Imagination]# ln -s /Universitas_Beyond_Imagination/WR1/Kemahasiswaan/Bina_karakter/program_bina_karakter.txt /WR1
[root@serveransible Universitas_Beyond_Imagination]#
```

```
_Imagination]# ln -s /Universitas_Beyond_Imagination/WR1/Kemahasiswaan/Bina_karakter/program_bina_karakter.txt /WR1
_Imagination]#
```

Penjelasan: command diatas merupakan soft link ke file program\_bina\_karakter.txt

```
[root@serveransible ~]# ln -s /Universitas_Beyond_Imagination/WR2/Keuangan/RKA_unit/RKA unit kemahasiswaan.txt /WR2
[root@serveransible ~]# ls -l /Universitas_Beyond_Imagination/WR2/Keuangan/RKA_unit/RKA unit kemahasiswaan.txt /WR2
ls: cannot access /Universitas_Beyond_Imagination/WR2/Keuangan/RKA unit/RKA unit kemahasiswaan.txt: No such file or directory
lrwxrwxrwx 1 root root 80 Sep 23 14:12 /WR2 -> 
[root@serveransible ~]#
```

Penjelasan: command diatas merupakan soft link ke file RKA\_unit\_kemahasiswaan.txt

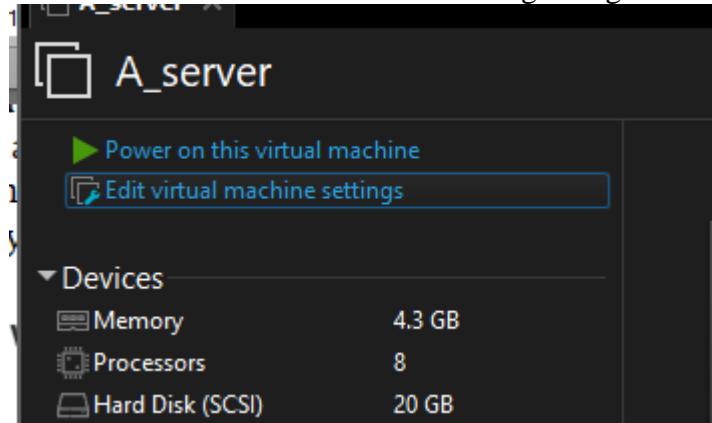
```
[root@serveransible ~]# ls -l /Universitas_Beyond_Imagination/WR3/Kerjasama/Kerjasama_dalam_dan_luar_negeri.txt /WR3
ls: cannot access /Universitas_Beyond_Imagination/WR3/Kerjasama/Kerjasama_dalam_dan_luar_negeri.txt: No such file or directory
lrwxrwxrwx 1 root root 91 Sep 23 14:25 /WR3 -> /Universitas_Beyond_Imagination/WR3/Kerjasama/Kerjasama_dalam_dan_luar_negeri.txt
[root@serveransible ~]#
```

Penjelasan: command diatas merupakan soft link ke file kerjasama\_dalam\_dan\_luar\_negeri.txt

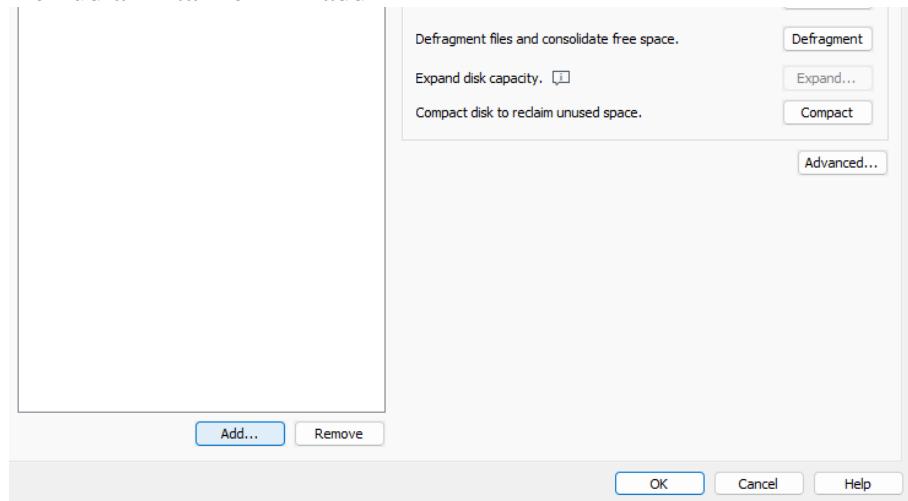
# BAB 6. FILESYSTEM AND LOGICAL VOLUME MANAGER – PART II

## 1. Menambahkan 6 storage baru dengan ukuran 256 MB

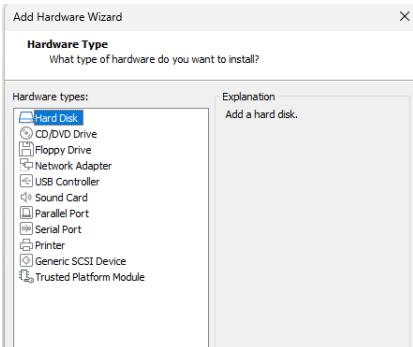
- Pada VMWare kita menambahkan storage dengan menekan edit virtual machine settings



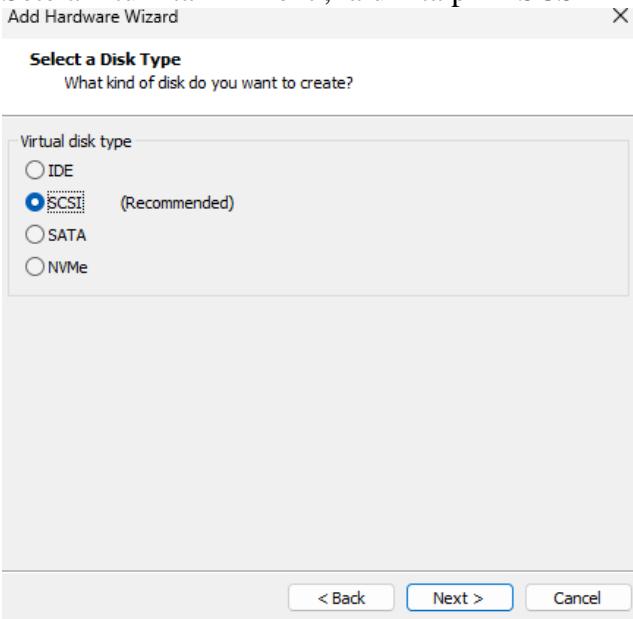
- Kemudian kita memilih add



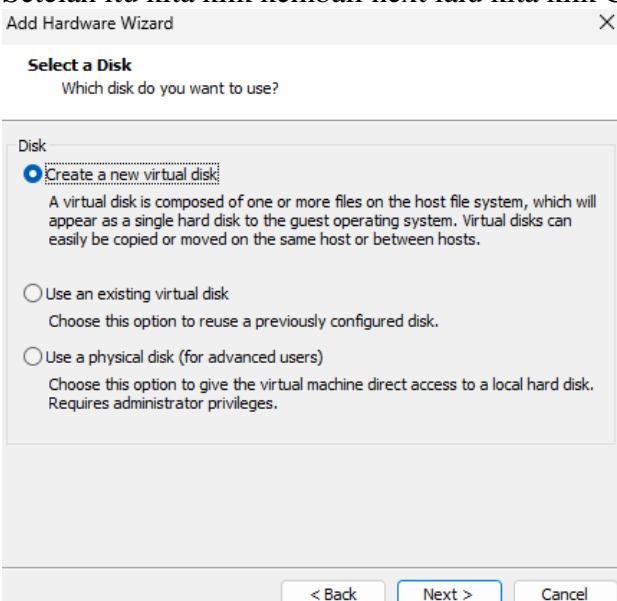
- Kemudian kita memilih hard disk



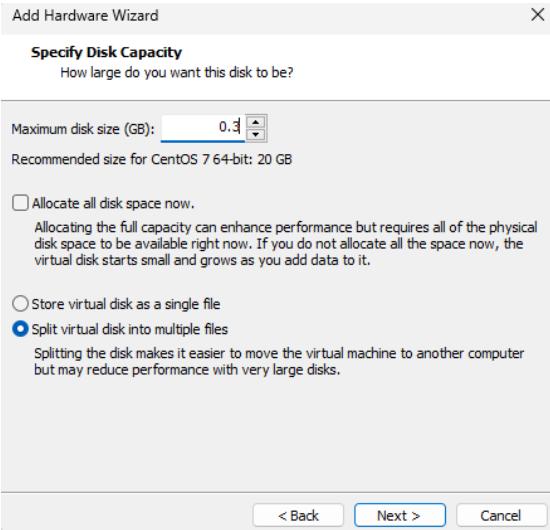
- Setelah itu kita klik next , lalu kita pilih SCSI



- Setelah itu kita klik kembali next lalu kita klik Create a new virtual disk



- Kemudian kita mengatur jumlah penyimpanannya lalu kita pilih split virtual disk into multiple files lalu kita pilih next.



- Lalu atur penamaanya lalu klik finish, ulangi langkah yang sama dari awal hingga terdapat 6 virtual disk baru.
- Lalu kita kembali ke edit virtual machine settings dan kita dapat melihat bahwa 6 virtual disk baru telah tercreate.

Device	Summary
Memory	4.3 GB
Processors	8
Hard Disk (SCSI)	20 GB
Hard Disk 3 (SCSI)	2 GB
Hard Disk 2 (SCSI)	2 GB
New Hard Disk (SCSI)	307 MB
New Hard Disk (SCSI)	307 MB
New Hard Disk (SCSI)	307 MB
New Hard Disk (SCSI)	307 MB
New Hard Disk (SCSI)	307 MB
CD/DVD (IDE)	Auto detect
Network Adapter	NAT
USB Controller	Present
Sound Card	Auto detect
Printer	Present
Display	Auto detect

Setelah keenam virtual disk yang baru telah ditambahkan pada mesin virtual, selanjutnya jalankan langkah-langkah berikut.

1. Setelah sistem operasi berjalan, buka aplikasi terminal.

2. Konfigurasi LVM dapat dieksekusi menggunakan akun root dengan perintah su –
3. Untuk mengetahui hard drive yang telah ditambahkan pada sistem operasi jalankan perintah fdisk -l. Misalnya, pada gambar berikut keenam hard drive yang baru ditunjukkan pada Disk /dev/sdb – Disk /dev/sdg.

```

Template Linux Exam X00-004 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminal

/dev/sda7      34512896   38707199   2097152   8e  Linux LVM

Disk /dev/sdb: 262 MB, 262144000 bytes, 512000 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdc: 262 MB, 262144000 bytes, 512000 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdd: 262 MB, 262144000 bytes, 512000 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sde: 262 MB, 262144000 bytes, 512000 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdf: 262 MB, 262144000 bytes, 512000 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

```

Output:

```

root@serveransible:~#
File Edit View Search Terminal Help
root@serveransible:~

Device Boot      Start        End     Blocks   Id  System
/dev/sd1       *          2048     616447    307200   83  Linux
/dev/sd2          616448    4810751   2097152   82  Linux swap / Solaris
/dev/sd3          4810752   41943039   18566144   83  Linux

Disk /dev/sdd: 321 MB, 321912832 bytes, 628736 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sde: 321 MB, 321912832 bytes, 628736 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdf: 321 MB, 321912832 bytes, 628736 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdg: 321 MB, 321912832 bytes, 628736 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk /dev/sdb: 2147 MB, 2147483648 bytes, 4194304 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x29c18014

```

4. Pilihlah 2 hard drive yang akan dikonversi menjadi physical volume sebagai penampung LVM, dengan menjalankan perintah pvcreate. Misalnya, seperti yang ditunjukkan pada gambar berikut:

```
# pvcreate /dev/sdb ; pvcreate /dev/sdc ; pvdisplay
```

Output:

Disini saya menggunakan hard drive /dev/sdd dan /dev/sde.

```
[root@serveransible ~]# pvcreate /dev/sdd ; pvcreate /dev/sde ; pvdisplay
Physical volume "/dev/sdd" successfully created.
Physical volume "/dev/sde" successfully created.
"/dev/sde" is a new physical volume of "307.00 MiB"
--- NEW Physical volume ---
PV Name          /dev/sde
VG Name
PV Size          307.00 MiB
Allocatable      NO
PE Size          0
Total PE         0
Free PE          0
Allocated PE     0
PV UUID          4X1mFf-CHyS-wMMA-7abM-iHpk-olKo-qAq02J

"/dev/sdd" is a new physical volume of "307.00 MiB"
--- NEW Physical volume ---
PV Name          /dev/sdd
VG Name
PV Size          307.00 MiB
Allocatable      NO
PE Size          0
Total PE         0
Free PE          0
Allocated PE     0
PV UUID          C4vUFc-qJYm-i4Ee-yvCi-n9tk-QJ3l-5LwrIV

[root@serveransible ~]#
```

5. Jika physical volume sudah terbentuk, buatlah volume group dengan nama home, seperti perintah berikut:

```
# vgcreate home /dev/sdb /dev/sdc ; vgdisplay
```

Output:

```
[root@serveransible ~]# vgcreate home /dev/sdd /dev/sde ; vgdisplay
Volume group "home" successfully created
--- Volume group ---
VG Name          home
System ID
Format          lvm2
Metadata Areas  2
Metadata Sequence No 1
VG Access        read/write
VG Status        resizable
MAX LV
Cur LV
Open LV
Max PV
Cur PV
Act PV
VG Size          608.00 MiB
PE Size          4.00 MiB
Total PE         152
Alloc PE / Size  0 / 0
Free PE / Size   152 / 608.00 MiB
VG UUID          7ZUga3-7V40-bLUF-KmKe-L2cR-KLS6-1VRrTz

[root@serveransible ~]# ■
```

6. Alokasikan 300 MB Logical Volume yang disebut dengan home\_directory dari volume group yang sudah dibuat, seperti perintah berikut:

```
# lvcreate -L 300M -n home_directory ; lvdisplay
```

Output:

```
[root@serveransible ~]# lvcreate -n home_directory -L 300M home
  Logical volume "home_directory" created.
[root@serveransible ~]# lvdisplay
--- Logical volume ---
  LV Path          /dev/home/home_directory
  LV Name          home_directory
  VG Name          home
  LV UUID          OVODD9-eZeh-uKFZ-Tfej-lyrn-IbwT-ieB0Cd
  LV Write Access  read/write
  LV Creation host, time serveransible, 2022-10-04 09:01:10 +0700
  LV Status        available
  # open           0
  LV Size          300.00 MiB
  Current LE       75
  Segments         1
  Allocation       inherit
  Read ahead sectors auto
  - currently set to 8192
  Block device     253:0
```

---

```
[root@serveransible ~]# █
```

7. Buatlah filesystem kemudian mount ke direktori yang baru, seperti perintah berikut:

```
# mkfs.ext4 /dev/home/home_directory ; mkdir home_directory
# mount /dev/home/home_directory /home_directory/
# lvdisplay
```

#### Output:

```
[root@serveransible ~]# mkfs.ext4 /dev/home/home_directory ; mkdir home_directory
mke2fs 1.42.9 (28-Dec-2013)
Filesystem label=
OS type: Linux
Block size=1024 (log=0)
Fragment size=1024 (log=0)
Stride=0 blocks, Stripe width=0 blocks
76912 inodes, 307200 blocks
15360 blocks (5.00%) reserved for the super user
First data block=1
Maximum filesystem blocks=33947648
38 block groups
8192 blocks per group, 8192 fragments per group
2024 inodes per group
Superblock backups stored on blocks:
  8193, 24577, 40961, 57345, 73729, 204801, 221185

Allocating group tables: done
Writing inode tables: done
Creating journal (8192 blocks): done
Writing superblocks and filesystem accounting information: done

[root@serveransible ~]# █
[...]
[root@serveransible ~]# mount /dev/home/home_directory/ /home_directory/
mount: /dev/mapper/home-home_directory is already mounted or /home_directory busy
      /dev/mapper/home-home_directory is already mounted on /home_directory
[root@serveransible ~]# █
```

```
[root@serveransible ~]# lvdisplay
  --- Logical volume ---
  LV Path          /dev/home/home_directory
  LV Name          home_directory
  VG Name          home
  LV UUID          OVODD9-eZeh-uKFZ-Tfej-lyrn-IbwT-ieB0Cd
  LV Write Access  read/write
  LV Creation host, time serveransible, 2022-10-04 09:01:10 +0700
  LV Status        available
  # open           0
  LV Size          300.00 MiB
  Current LE      75
  Segments         1
  Allocation       inherit
  Read ahead sectors auto
  - currently set to 8192
  Block device    253:0
```

8. Resize logical volume `home_directory` menjadi 400MB kemudian tampilkan hasilnya dengan perintah `df -h`.

```
# lvresize -r -L 400M /dev/home/home_directory
# df -h
```

Output:

```
[root@serveransible ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/devtmpfs        2.0G   0    2.0G  0% /dev
tmpfs           2.0G  12K  2.0G  1% /dev/shm
tmpfs           2.0G  13M  2.0G  1% /run
tmpfs           2.0G   0    2.0G  0% /sys/fs/cgroup
/dev/sda3        18G  18G  9.4M 100% /
/dev/sda1       297M 294M  2.9M 100% /boot
/dev/md0         2.0G  6.1M  1.9G  1% /mnt/raid1
/dev/md1         2.0G  6.1M  1.9G  1% /mnt/raid1_tambahan
/dev/mapper/home-home_directory 481M  2.3M 449M  1% /home_directory
tmpfs           407M  20K  407M  1% /run/user/0
[root@serveransible ~]#
```

9. Jika kapasitas masih dibutuhkan, tambahkan 50MB, seperti perintah berikut:

```
# lvresize -r -L +50M /dev/home/home_directory
# df -h
```

Output:

```
[root@serveransible ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/devtmpfs        2.0G   0    2.0G  0% /dev
tmpfs           2.0G  12K  2.0G  1% /dev/shm
tmpfs           2.0G  13M  2.0G  1% /run
tmpfs           2.0G   0    2.0G  0% /sys/fs/cgroup
/dev/sda3        18G  18G  9.4M 100% /
/dev/sda1       297M 294M  2.9M 100% /boot
/dev/md0         2.0G  6.1M  1.9G  1% /mnt/raid1
/dev/md1         2.0G  6.1M  1.9G  1% /mnt/raid1_tambahan
/dev/mapper/home-home_directory 481M  2.3M 449M  1% /home_directory
tmpfs           407M  20K  407M  1% /run/user/0
[root@serveransible ~]#
```

10. Tambahkan logical volume `home_directory` pada file `/etc/fstab`, sehingga logical volume dapat di-mount ketika sistem diaktifkan. Pada line yang sama terapkan kuota untuk user dan group, sehingga

penggunaan storage dapat dibatasi, seperti perintah berikut:

```
/dev/md1          /home_directory ext4 defaults,usrquota,grpquota      1 2
```

Output:

Kemudian kita masuk ke /etc/fstab dengan command nano /etc/fstab

Lalu kita tambahkan isi filenya seperti gambar dibawah

```
/dev/md1          /mnt/raid1_tambahan ext4 defaults 0 0
/dev/home/home_directory /home_directory ext4 defaults,usrquota,grpquota      1 2
```

11. Selanjutnya, untuk mengaktifkan pengaturan kuota yang telah didefinisikan pada file /etc/fstab, maka jalankan perintah berikut:

```
quotacheck -cug /home_directory
```

Jika kuota berhasil diaktifkan, maka akan terbentuk dua file yaitu aquota.user dan aquota.group.

```
# cd /home_directory/
# ls -l
total 30
-rw----- 1 root root 7168 Sep 29 06:51 aquota.group
-rw----- 1 root root 7168 Sep 29 06:51 aquota.user
```

Output:

```
[root@serveransible ~]# quotacheck -cug /home_directory
[root@serveransible ~]# 
[root@serveransible ~]# cd /home_directory/
[root@serveransible home_directory]# ls -l
total 24
-rw----- 1 root root 6144 Oct  4 10:04 aquota.group
-rw----- 1 root root 6144 Oct  4 10:04 aquota.user
drwx----- 2 root root 12288 Oct  4 09:23 lost+found
[root@serveransible home_directory]#
```

12. Selanjutnya, jika user ditambahkan pada sistem, maka home direktori akan diarahkan ke logical volume /home\_directory. Hal ini mungkin dilakukan dengan mengubah file /etc/default/useradd. Ubahlah variable HOME mengarah ke direktori home yang baru yaitu /home\_directory. Uji cobalah apakah ketika user yang baru ditambahkan maka direktori home yang terbentuk ada pada /home\_directory.

Output:

Ketik mengetik nano /etc/default/useradd lalu mengeditnya seperti dibawah

```
GNU nano 2.3.1                                         File: /etc/default/useradd

# useradd defaults file
GROUP=100
HOME=/home_directory
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes
```

```

File Edit View Search Terminal Help
GNU nano 2.3.1          File: /etc/default/useradd

# useradd defaults file
GROUP=100
HOME=/home_directory
INACTIVE=-1
EXPIRE=
SHELL=/bin/bash
SKEL=/etc/skel
CREATE_MAIL_SPOOL=yes

```

13. Kuota dapat diterapkan pada single user ataupun group dengan menggunakan perintah edquota username/group. Buatlah sebuah user dengan username test1, seharusnya direktori home yang tercipta berada pada /home\_directory, seperti berikut:

```
# pwd
/home_directory/test1
```

Output:

```
[root@serveransible ~]# cd /home_directory
[root@serveransible home_directory]# ls
aquota.group  aquota.user  lost+found  test1
[root@serveransible home_directory]#
```

14. Kemudian terapkan kuota untuk user test1 dengan menjalankan perintah edquota *username* misalnya edquota test1. Ketika perintah ini dijalankan kuota untuk user test1 dapat dikonfigurasi, tambahkan 55KB pada bagian hard, seperti gambar berikut:

	blocks	soft	hard	inodes	soft	hard
1 Disk quotas for user test1 (uid 1003):						
2 Filesystem	0	0	0	0	0	0
3 /dev/sda6	0	0	0	0	0	0
4 /dev/mapper/home-home_directory	0	0	55000	0	0	0

Output:

Kita mengetik command edquota test1 dan menambahkan baris seperti dibawah

```

File Edit View Search Terminal Help
Disk quotas for user test1 (uid 1044):
  Filesystem    blocks    soft    hard    inodes    soft    hard
  /dev/mapper/home-home_directory      0        0    55000      0        0        0
  ~
  ~
  ~

```

15. Lakukanlah uji coba untuk memastikan bahwa user hanya dapat menggunakan direktori home sesuai dengan kuota yang telah ditetapkan dengan menambahkan file pada direktori home user test1. Selanjutnya, untuk memeriksa kuota yang telah digunakan user, jalankan perintah seperti pada gambar berikut:

```
# quota test1
Disk quotas for user test1 (uid 1003):
  Filesystem  blocks  quota  limit  grace  files  quota  limit  grace
  /dev/mapper/home-home_directory      55001*    0    55000            1        0        0
```

Output:

```
[root@serveransible ~]# quota test1
Disk quotas for user test1 (uid 1044): no limited resources used
[root@serveransible ~]#
```

16. Untuk mendapatkan laporan penggunaan kuota untuk seluruh user dapat dilakukan dengan menjalankan perintah repquota -a, seperti gambar berikut:

```
[root@hostansible ~]# repquota -a
*** Report for user quotas on device /dev/mapper/home-home_directory
Block grace time: 7days; Inode grace time: 7days
      Block limits          File limits
User    used   soft   hard grace   used   soft   hard grace
-----
root    --     13     0     0           2     0     0
test1   --     10     0     55          5     0     0
```

17. Jika kapasitas pada sebuah volume group perlu ditambah, maka dapat dilakukan dengan menjalankan perintah vgextend. Sebelum menambah kapasitas dari hard drive lainnya, harus dipastikan bahwa hard drive tersebut belum dialokasikan menjadi filesystem untuk program lain, dengan perintah fdisk -l. Misalnya, logical volume home yang dialokasikan untuk direktori home kapasitasnya perlu ditambah dan terdapat hard drive yang free pada /dev/sdg, maka perintah yang akan dijalankan seperti berikut:

```
# vgextend home /dev/sdg
```

Periksalah apakah kapasitas untuk logical volume home sudah bertambah.

Output:

```
[root@serveransible ~]# vgextend home /dev/sdf
Physical volume "/dev/sdf" successfully created.
Volume group "home" successfully extended
[root@serveransible ~]#
[root@serveransible ~]# vgdisplay
--- Volume group ---
VG Name          home
System ID
Format          lvm2
Metadata Areas  3
Metadata Sequence No  6
VG Access       read/write
VG Status        resizable
MAX LV
Cur LV
Open LV
Max PV
Cur PV
Act PV
VG Size         912.00 MiB
PE Size          4.00 MiB
Total PE         228
Alloc PE / Size 126 / 504.00 MiB
Free  PE / Size 102 / 408.00 MiB
VG UUID          7ZUga3-7V40-bLUF-KmKe-L2cR-KLS6-1VRrTz
```

18. Selanjutnya, hard drive (physical volume) dapat dipisahkan dari volume group, namun harus dipastikan terlebih dahulu bahwa physical volume tersebut tidak sedang digunakan oleh logical volume dengan menjalankan perintah pvdisplay. Misalnya, hapuslah physical volume /dev/sdg yang telah dialokasikan pada volume group home, dengan menjalankan perintah berikut:

```
# vgreduce home /dev/sdg
```

Output:

```
[root@serveransible ~]# vgreduce home /dev/sdf
  Removed "/dev/sdf" from volume group "home"
[root@serveransible ~]#
[root@serveransible ~]# vgdiskl
--- Volume group ---
VG Name          home
System ID
Format           lvm2
Metadata Areas   2
Metadata Sequence No 7
VG Access        read/write
VG Status        resizable
MAX LV           0
Cur LV           1
Open LV           1
Max PV           0
Cur PV           2
Act PV           2
VG Size          608.00 MiB
PE Size          4.00 MiB
Total PE         152
Alloc PE / Size 126 / 504.00 MiB
Free PE / Size  26 / 104.00 MiB
VG UUID          7ZUga3-7V40-bLUF-KmKe-L2cR-KLS6-1VRrTz
```

19. Jalankanlah kembali instruksi pada nomor 17, kemudian bagilah volume group dengan membentuk volume group yang baru. Namun sebelumnya dipastikan bahwa logical volume tidak dapat dibagi pada volume group yang berbeda. Setiap logical volume yang sudah terbentuk harus berada pada physical volume yang telah digabungkan pada sebuah volume group apakah volume group yang baru atau volume group yang lama. Misalnya, membagi volume group yang baru dengan volume group yang telah ada, seperti perintah berikut:

```
# vgsplit home new_home /dev/sdg
```

Output:

- Membuat pyhsical volume baru dengan /dev/sdh /dev/sdi

```
[root@serveransible ~]# pvcreate /dev/sdh ; pvcreate /dev/sdi ; vgdiskl
Physical volume "/dev/sdh" successfully created.
Physical volume "/dev/sdi" successfully created.
--- Physical volume ---
```

- Membuat volume group baru

```
[root@serveransible ~]# vgcreate new_home /dev/sdh /dev/sdi ; vgdiskl
Volume group "new_home" successfully created
--- Volume group ---
VG Name          new_home
System ID
Format           lvm2
Metadata Areas   2
Metadata Sequence No 1
VG Access        read/write
VG Status        resizable
MAX LV           0
Cur LV           0
Open LV           0
Max PV           0
Cur PV           2
Act PV           2
VG Size          608.00 MiB
PE Size          4.00 MiB
Total PE         152
Alloc PE / Size 0 / 0
Free PE / Size  152 / 608.00 MiB
VG UUID          687iSa-Vmqc-TLBx-gN9K-0RQh-kUCi-iHE9LG
```

- Kemudian kita membagi /dev/sdf ke home dan new\_home

```
[root@serveransible ~]# vgsplit home new_home /dev/sdg
Logical volumes in "home" must be inactive
[root@serveransible ~]# vgchange -an home
Logical volume home/home_directory contains a filesystem in use.
Can't deactivate volume group "home" with 1 open logical volume(s)
[root@serveransible ~]# █
```

## BAB 7. SAMBA DOMAIN CONTROLLER

### A. Instalasi Samba pada CentOS

Langkah-langkah untuk melakukan instalasi Samba pada CentOS:

Instal samba-client dengan mengeksekusi perintah berikut

**yum install samba samba-client**

Output:

```
[root@localhost ~]# yum -y install samba samba-client
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
epel/x86_64/metalink
 * base: mr.heru.id
 * epel: mirror.sabay.com.kh
 * extras: mr.heru.id
 * remi-php81: ftp.riken.jp
 * remi-safe: ftp.riken.jp
 * updates: mirror.vodien.com
| 4.1 kB  00:00:00
----
```

Setelah proses instalasi selesai, jalankan servis samba dan jalankan perintah yang akan menjalankan

servis secara otomatis pada saat system boot. Eksekusi perintah berikut:

```
# systemctl start smb.service
# systemctl start nmb.service
# systemctl enable smb.service
# systemctl enable nmb.service
```

Output:

```
[root@localhost ~]# systemctl start smb.service
[root@localhost ~]# systemctl start nmb.service
[root@localhost ~]# systemctl enable smb.service
Created symlink from /etc/systemd/system/multi-user.target.wants/smb.service to /usr/lib/systemd/system/smb.service.
[root@localhost ~]# systemctl enable nmb.service
Created symlink from /etc/systemd/system/multi-user.target.wants/nmb.service to /usr/lib/systemd/system/nmb.service.
[root@localhost ~]#
```

Servis smbd menyediakan layanan berbagi file (sharing file) dan layanan cetak (printing) yang aktif pada port TCP 139 dan 445. Selanjutnya, servis nmbd menyediakan layanan NETBIOS melalui layanan penamaan IP terhadap klien yang aktif pada port UDP 137.

Setelah Samba berhasil maka seluruh port yang penting harus terbuka, dengan menjalankan perintah berikut

```
firewall-cmd --permanent --zone=public --add-service=samba
firewall-cmd --zone=public --add-service=samba
```

Output:

```
[root@localhost ~]# firewall-cmd --permanent --zone=public --add-service=samba
success
[root@localhost ~]# firewall-cmd --zone=public --add-service=samba
success
[root@localhost ~]#
```

### Konfigurasi Samba

Studi kasus yang digunakan pada praktikum ini adalah modul Pengelolaan User dan Group pada Linux. Pada modul ini diberikan satu contoh penerapan Samba untuk divisi Finance, dengan langkah-langkah sebagai berikut:

Tambahkan group Finance dengan menggunakan perintah groupadd.

```
# groupadd finance
```

Buatlah folder data yang kemudian setelahnya didalam folder /data, sama seperti pada modul Pengelolaan User dan Group. Langkah ini bisa dilewatkan jika folder telah dipersiapkan pada praktikum sebelumnya.

```
# mkdir data
```

Tambahkan folder finance dibawah folder data dengan path folder /data/finance.

```
# mkdir finance
```

Output untuk langkah 1-3

```
[root@localhost ~]# mkdir data
[root@localhost ~]# groupadd finance
[root@localhost ~]# cd data
[root@localhost data]# mkdir financi
[root@localhost data]# mkdir finance
```

Ketiga perintah berikut dijalankan untuk mengubah mode akses folder /data/finance menjadi 770 secara rekursif. Kemudian ubah konteks SELinux security menjadi samba\_share\_t pada folder /data/finance. Selanjutnya, ubah hak kepemilikan folder /data/finance dengan root sebagai pemilik folder dan juga group finance.

```
chmod -R 770 /data/finance/
chcon -t samba_share_t /data/finance
chown -R root:finance /data/finance
```

Output:

```
[root@localhost ~]# chmod -R 770 data/finance
[root@localhost ~]# chcon -t samba_share_t data/finance
[root@localhost ~]# chown =R root:finance data/finance
chown: invalid user: '=R'
[root@localhost ~]# chown -R root:finance data/finance
[root@localhost ~]#
```

Berikutnya, dari user yang sudah diciptakan pada modul praktikum sebelumnya, contohnya bylardo dari bagian Finance, maka user bylardo harus juga ditetapkan sebagai user samba. Smbpasswd adalah perintah yang digunakan untuk menciptakan user samba seperti perintah berikut.

```
# smbpasswd -a bylardo
```

New SMB password:

Retype new SMB password:

Added user bylardo.

```
[root@localhost ~]# smbpasswd -a Bylardo
New SMB password:
Retype new SMB password:
Added user Bylardo.
[root@localhost ~]#
```

Setelah akun bylardo ditetapkan menjadi user samba, selanjutnya user bylardo diaktifkan dengan menggunakan perintah smbpasswd dengan argumen -e.

```
# smbpasswd -e bylardo
```

Enabled user bylardo.

Output:

```
[root@localhost ~]# smbpasswd -e Bylardo
Enabled user Bylardo.
[root@localhost ~]#
```

Setelah itu, lakukan konfigurasi pada file /etc/samba/smb.conf. Sebelumnya, sangat dihindari untuk memodifikasi file konfigurasi yang original. Oleh karena itu sebelum menambahkan konfigurasi maka buatkan file cadangan.

Tambahkan konfigurasi folder berbagi pada file konfigurasi /etc/samba/smb.conf seperti konfigurasi berikut.

[Finances] → nama folder berbagi yang akan terlihat

Path → hierarki folder yang akan dibagi melalui servis samba, dalam hal ini adalah /data/finance.

Browseable → folder berbagi dapat diakses dan dieksplor

Writable → isi folder dapat ditulis dengan menambahkan folder atau file.

Read only → folder tidak dibatasi hanya dengan mode baca

Valid user → user valid yang dapat mengakses folder berbagi.

```
[Finances]
    path = /data/finance
    browseable = yes
    writeable = yes
    read only = no
    valid users = @finance
```

```
File Edit View Search Terminal Help
GNU nano 2.3.1          File: /etc/samba/smb.conf

path = /var/lib/samba/drivers
write list = @printadmin root
force group = @printadmin
create mask = 0664
directory mask = 0775
[finance]
    path = data/finance
    browseable = yes
    writeable = yes
    read only = no
    valid users = @finance
```

Jika ada perubahan yang diterapkan pada file konfigurasi samba, maka diakhiri dengan mengaktifkan kembali servis dari samba dan netbios, seperti perintah berikut.

```
# systemctl restart smb.service
# systemctl restart nmb.service
```

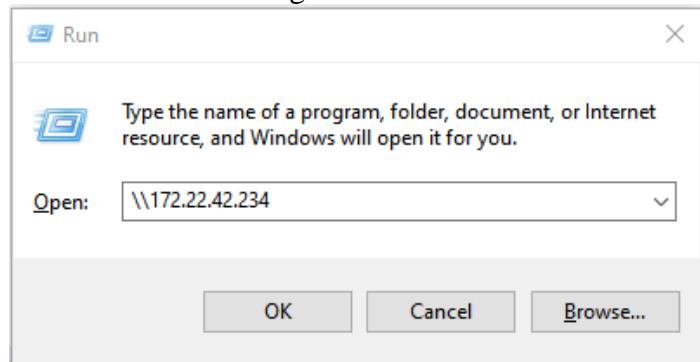
Output:

```
[root@localhost samba]# systemctl restart smb.service
[root@localhost samba]# systemctl restart nmb.service
```

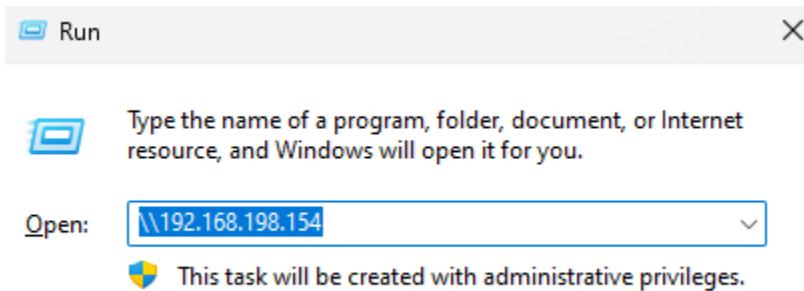
### Akses samba dari Windows

Setelah Samba dikonfigurasi maka dilakukan pengujian akses terhadap folder berbagi yang dalam hal ini adalah folder /data/finance. Langkah-langkah berikut untuk mengakses folder berbagi melalui sistem operasi Windows.

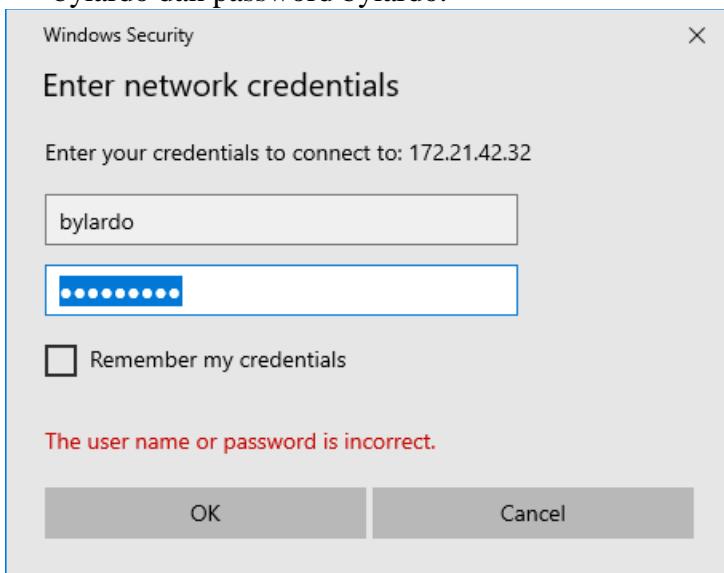
Akseslah folder berbagi melalui IP Address server samba, sebagai contoh seperti gambar berikut.



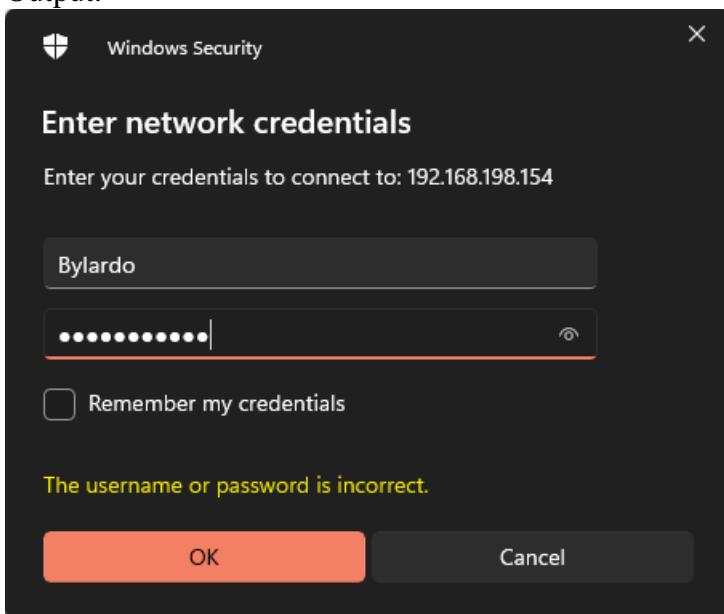
Output:



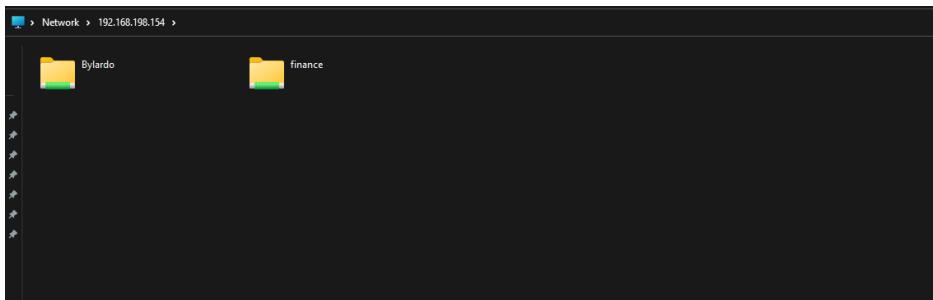
Jika server samba dapat diakses maka muncul tampilan berikut dengan cara memberikan username bylardo dan password bylardo.



Output:



Jika akun bylardo berhasil login, maka folder Finance akan muncul pada folder berbagai sesuai dengan yang telah dikonfigurasi pada konfigurasi samba.

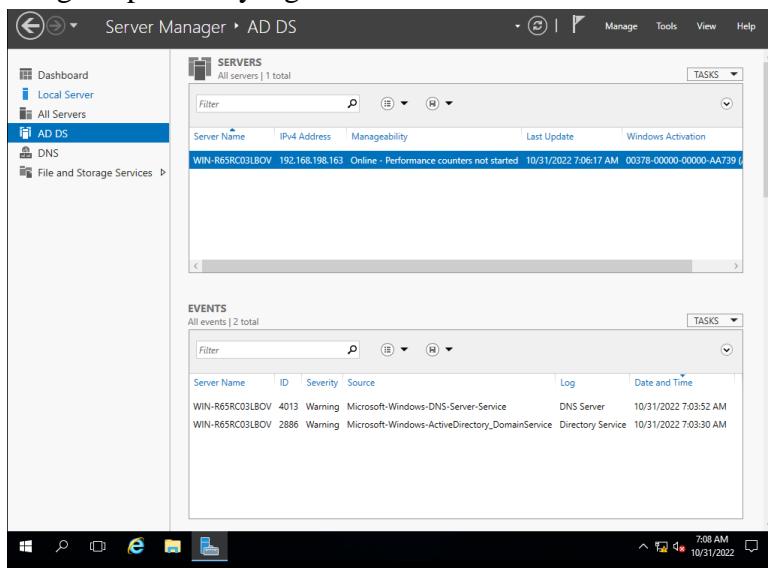


Selanjutnya, dilakukan uji coba dengan membuat folder yang baru.

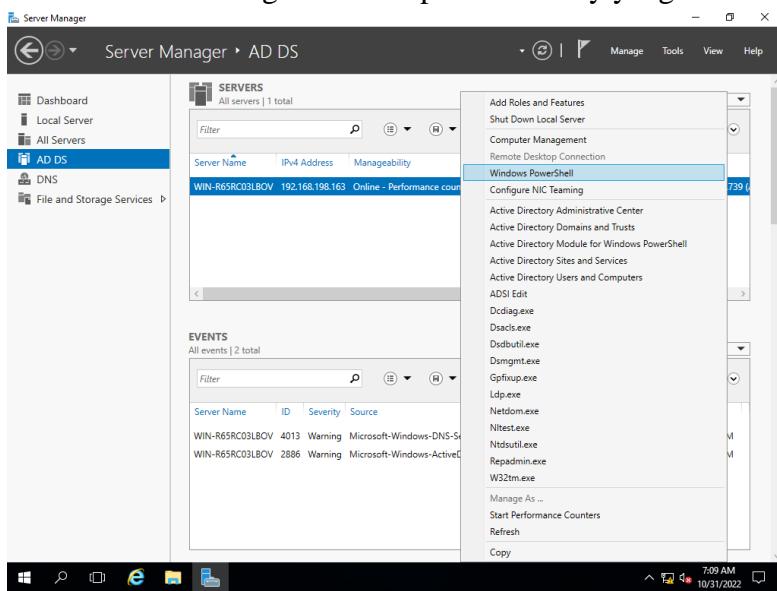
Name	Date modified	Type	Size
test	30/11/2022 20:03	File folder	

# BAB 8. PENGELOLAAN OBJEK PADA ACTIVE DIRECTORY MENGGUNAKAN POWERSHELL

1. Langkah pertama yang dilakukan adalah membuka windows server dan membuka AD DS



2. Setelah itu kita mengklik kanan pada directory yang aktif dan membuka powershell



3. Kemudian kita menjalankan perintah seperti dibawah

```

Administrator: Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> Get-Command New-ADUser -Syntax
New-ADUser [-Name <string>] [-WhatIf] [-Confirm] [-AccountExpirationDate <datetime>] [-AccountDisabled <bool>] [-Acc
ountPassword <securestring>] [-AllowReversiblePasswordEncryption <bool>] [-AuthenticationPolicy <AuthenticationPolicy>
] [-AuthenticationPolicyId <ADAuthenticationPolicyId>] [-AuthType <AuthType>] [-CannotChangePassword <bool>] [-Ce
rtificates <x509Certificate>] [-ChangePasswordAtLogon <bool>] [-City <string>] [-Company <string>] [-CompoundIdentitySu
pport <bool>] [-Country <string>] [-Department <string>] [-EmployeeAddress <string>] [-EmployeeID <string>] [-EmployeeNumber <string>] [-Enab
led <bool>] [-Fax <string>] [-GivenName <string>] [-HomeDirectory <string>] [-HomeDrive <string>] [-HomePage <string>] [-H
omePhone <string>] [-Initials <string>] [-Instance <string>] [-kerberosEncryptionType <string>] [-Loca
tion <string>] [-Manager <string>] [-MobilePhone <string>] [-Office <string>] [-OfficePhone <string>] [-OrgUnit <string>] [-OrgUnit
Path <string>] [-OtherAttributes <hashtable>] [-OtherName <string>] [-PassThru] [-PasswordNeverExpires <bool>] [-Passwo
rdNotRequired <bool>] [-Path <string>] [-POBox <string>] [-PostalCode <string>] [-PrincipalsAllowedToDelegateToAccount <
ADPrincipal[]>] [-ProfilePath <string>] [-ScriptPath <string>] [-SamAccountName <string>] [-Server <string>] [-SmartCardLogonRequired <bool>] [-State <string>] [-StreetAddress <string>] [-Surname <string>
] [-Title <string>] [-TrustedForDelegation <bool>] [-UserPrincipalName <string>] [<CommonParameters>]
PS C:\Users\Administrator>

```

- Kemudian kita bisa mencoba menjalankan command seperti dibawah

```

PS C:\Windows\system32> New-ADUser -Name "Toni Spark" -GivenName "Toni" -Surname "Spark" -SamAccount
Name "tspark" -UserPrincipalName tspark@hogwart.edu -Path "OU=IT,DC=Hogwart,DC=edu" -AccountPasswo
rd (Read-Host -AsSecureString "Type Password for user") -Enabled $true
Type Password for User: *****
PS C:\Windows\system32>

```

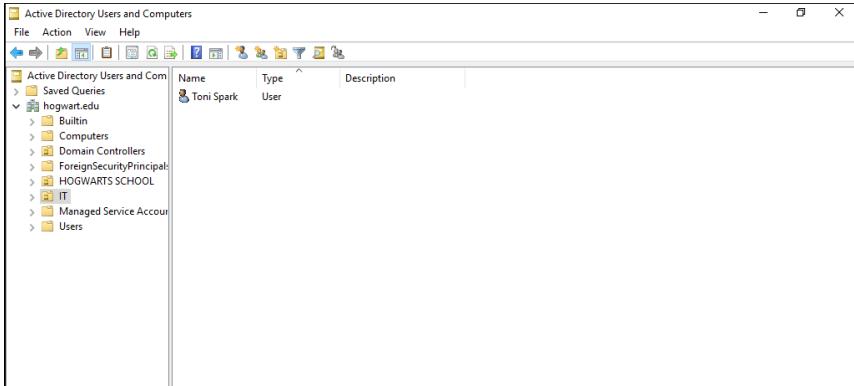
- Setelah itu kita bisa memeriksa user yang telah kita buat dengan command seperti dibawah

```

PS C:\Windows\system32> Get-ADUser tspark
DistinguishedName : CN=Toni Spark,OU=IT,DC=hogwart,DC=edu
Enabled : True
GivenName : Toni
Name : Toni Spark
ObjectClass : user
ObjectGUID : 6c68beec-2ef3-43b3-89fd-82dd0d078502
SamAccountName : tspark
SID : S-1-5-21-1934861792-524307133-28930124-1139
Surname : Spark
UserPrincipalName : tspark@hogwart.edu

```

Kita dapat melihatnya pada active directory



- Kita akan mencoba menambahkan user menggunakan file .csv . yang pertama kita lakukan adalah menyiapkan file berkextenstio .csv

Name	GivenName	Surname	SamAccountName	UserPrincipalName	Path
Coba1User1	Coba1	User1	cuser1	cuser1@hogwart.edu	OU=IT,DC=hogwart,DC=edu
Coba2User2	Coba2	User2	cuser2	cuser2@hogwart.edu	OU=IT,DC=hogwart,DC=edu
Coba3User3	Coba3	User3	cuser3	cuser3@hogwart.edu	OU=IT,DC=hogwart,DC=edu
Coba4User4	Coba4	User4	cuser4	cuser4@hogwart.edu	OU=IT,DC=hogwart,DC=edu
Coba5User5	Coba5	User5	cuser5	cuser5@hogwart.edu	OU=IT,DC=hogwart,DC=edu
Coba6User6	Coba6	User6	cuser6	cuser6@hogwart.edu	OU=IT,DC=hogwart,DC=edu
Coba7User7	Coba7	User7	cuser7	cuser7@hogwart.edu	OU=IT,DC=hogwart,DC=edu
Coba8User8	Coba8	User8	cuser8	cuser8@hogwart.edu	OU=IT,DC=hogwart,DC=edu
Coba9User9	Coba9	User9	cuser9	cuser9@hogwart.edu	OU=IT,DC=hogwart,DC=edu
Coba10User10	Coba10	User10	cuser10	cuser10@hogwart.edu	OU=IT,DC=hogwart,DC=edu

Kemudian kita mengeksekusi command seperti dibawah:

```
Import-Csv "C:\Users\Administrator\Desktop\coba1.csv" | ForEach-Object {
    $upn = $_.SamAccountName + "hogwart.edu"
    New-ADUser -Name $_."Name" `-
        -GivenName $_."GivenName" `-
        -Surname $_."Surname" `-
        -SamAccountName $_."SamAccountName" `-
        -UserPrincipalName $upn `-
        -Path $_."Path" `-
        -AccountPassword (ConvertTo-SecureString "Pa$$word_2018" -AsPlainText -Force) -Enabled $true
}
```

Output:

Name	Type	Description
Toni Spark	User	
Coba1User1	User	
Coba2User2	User	
Coba3User3	User	
Coba4User4	User	
Coba5User5	User	
Coba6User6	User	
Coba7User7	User	
Coba8User8	User	
Coba9User9	User	
Coba10User10	User	

7. Kemudian kita boleh mencoba memodifikasi data dari user dengan command seperti dibawah

```
Set-ADUser Coba1 -OfficePhone "1234567890" -City "London"
```

Output:

```
PS C:\Users\Administrator> Set-ADUser Cobal -OfficePhone "1234567890" -City "London"
```

```
PS C:\Users\Administrator>
```

8. Kemudian kita akan mencoba mengganti atribut dari user menggunakan command seperti dibawah

```
1 Get-AdUser -Filter * -SearchBase 'OU=IT,DC=hogwart,DC=edu' | Set-AdUser -City "London"
```

Output:

```
PS C:\Users\Administrator> Get-AdUser -Filter * -SearchBase 'OU=IT,DC=hogwart,DC=edu' | Set-AdUser -City "London"
```

```
PS C:\Users\Administrator>
```

Kita juga bisa mengubah atribut dengan command seperti dibawah

```
PS C:\Users\Administrator> Get-AdUser -Filter {City -like "London"} | Set-ADUser -City "Manchester"
```

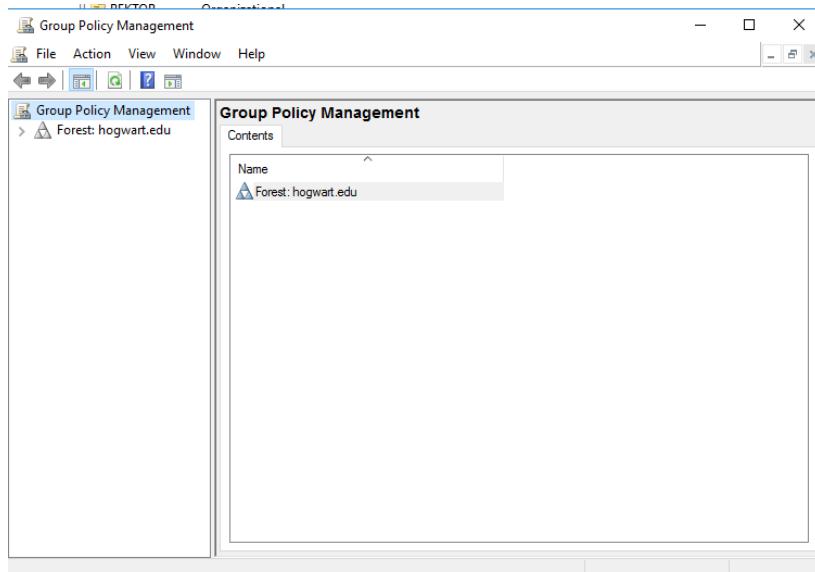
```
PS C:\Users\Administrator>
```

# BAB 9. MANAGING GROUP POLICY & ADVANCED AD MANAGEMENT WITH POWERSHELL

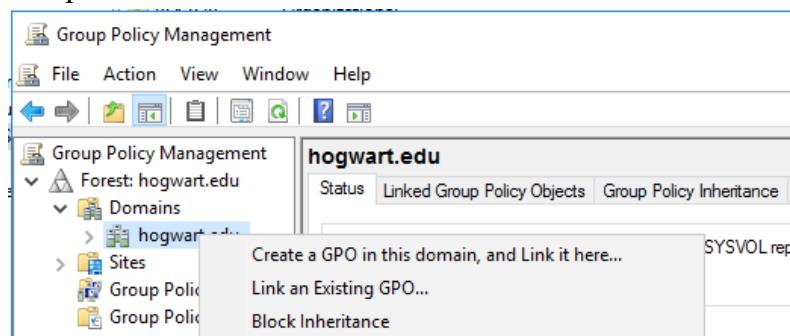
## A. Pengelolaan Group Policy pada AD

Untuk menciptakan Group Policy Object(GPO) pada sebuah AD dapat dilakukan melalui langkah-langkah berikut:

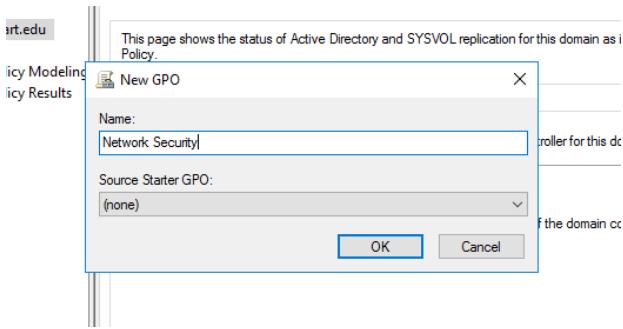
1. Buka fitur Group Policy Management melalui server manager. Melalui GP Management, dapat ditetapkan berbagai jenis policy kepada Organizational Unit(OU).



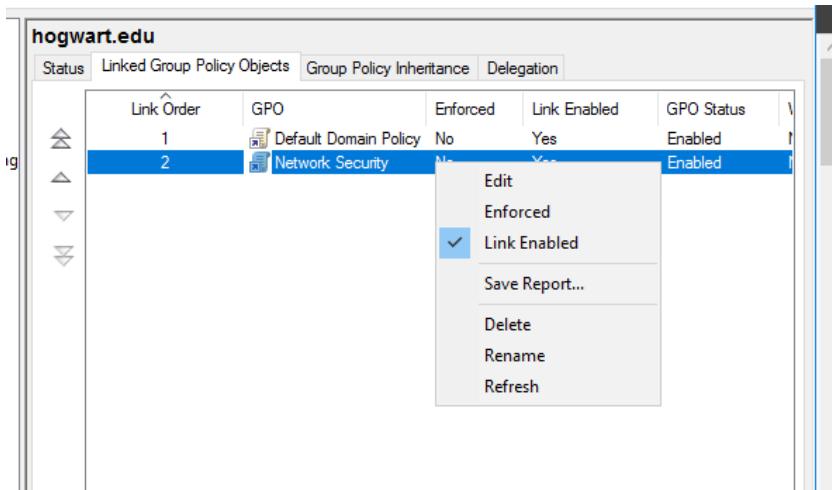
2. Setelah fitur GP management terbuka, selanjutkan klik kanan pada nama domain dan klik pilihan “create GPO in this domain and link here”.



3. Selanjutnya, berilah nama GPO. Contohnya adalah “Network Security”, kemudian klik OK.

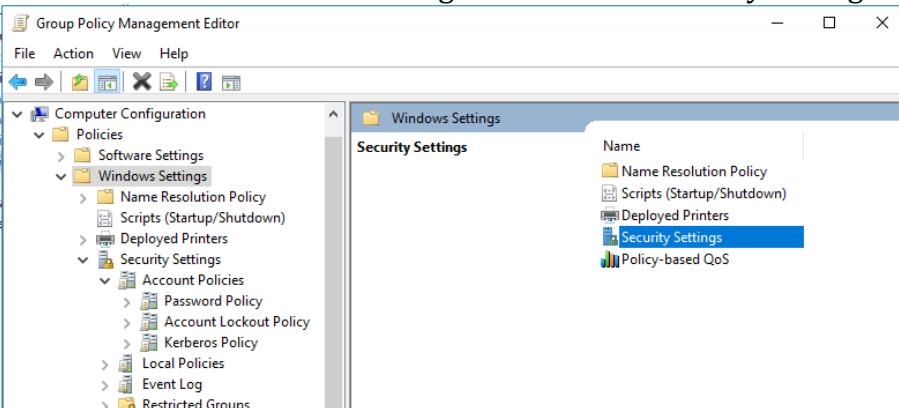


4. Setelah itu akan terbentuk sebuah GPO yang baru. Kemudian, klik kanan pada GPO Network Security dan klik edit.

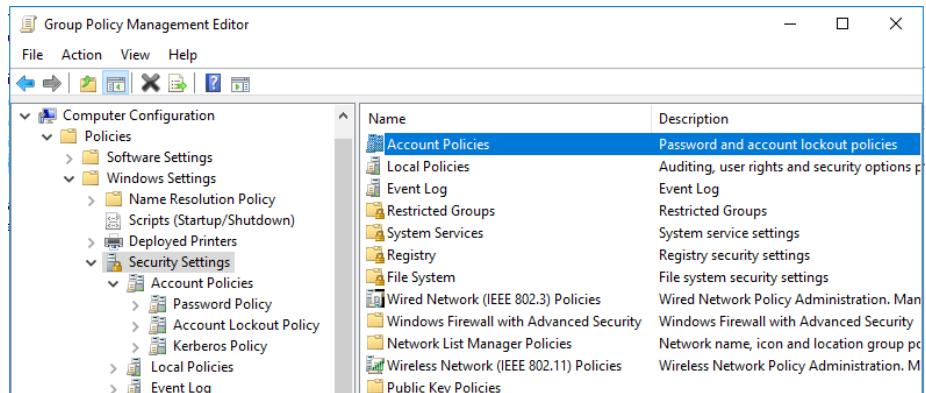


5. Selanjutnya, GP akan ditentukan. Sebagai contoh sederhana, pada Computer Configuration akan ditetapkan GPO pada bagian Windows Settings kemudian pilih Security Settings. Pilihlah Account Policies, kemudian Klik Password Policies. Pada bagian layar sebelah kanan akan ditampilkan jenis-jenis policy yang dapat dimodifikasi. Pilihlah salah satu dari jenis policy tersebut dan tentukan policy yang Anda tetapkan.

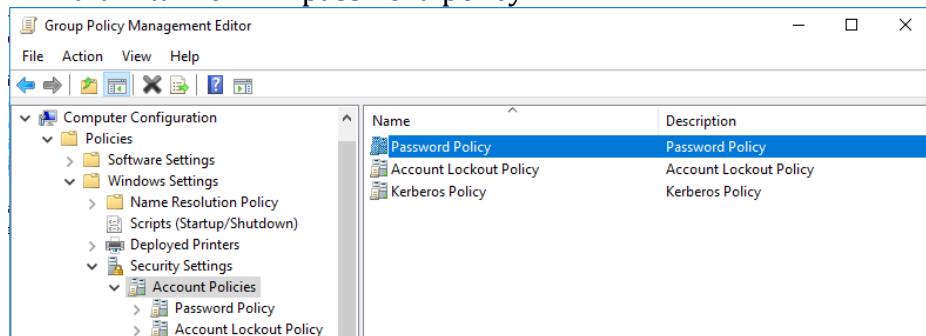
- Kita memilih windows settings dan memilih security settings



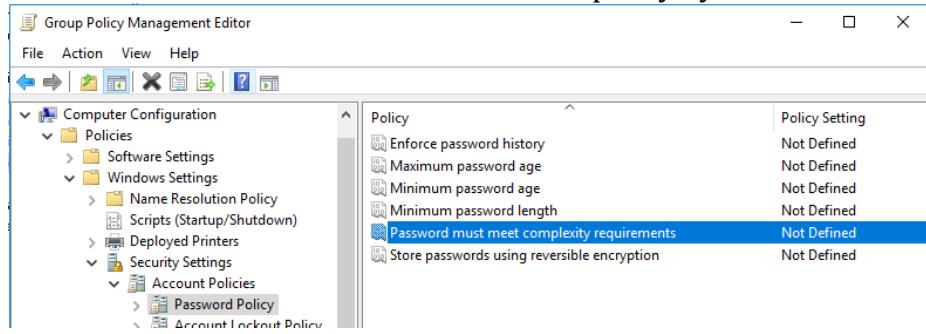
- Setelah masuk, kita memilih account polices



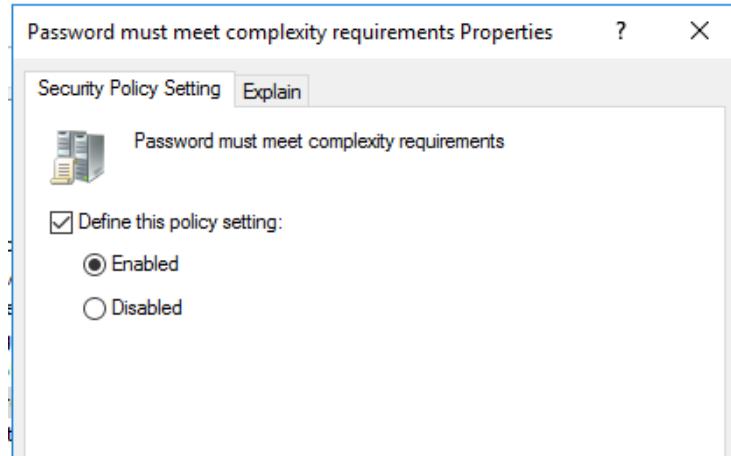
- Lalu kita memilih password policy



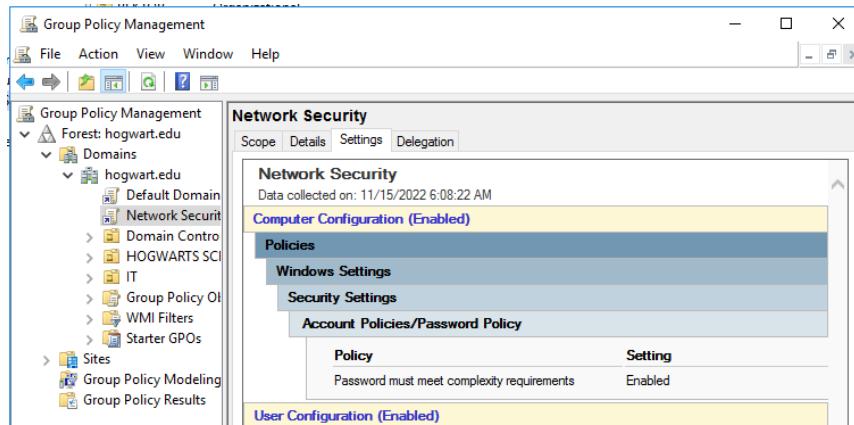
- Setelah itu kita memilih salah satu dari policy nya



- Kemudian kita mengklik dua kali kemudian mencentang checkbox dan memilih enabled



6. Untuk memverifikasi bahwa GPO telah berhasil ditetapkan maka klik GPO yang telah Anda modifikasi, lalu klik Settings kemudian periksa apakah policy telah diaktifkan.

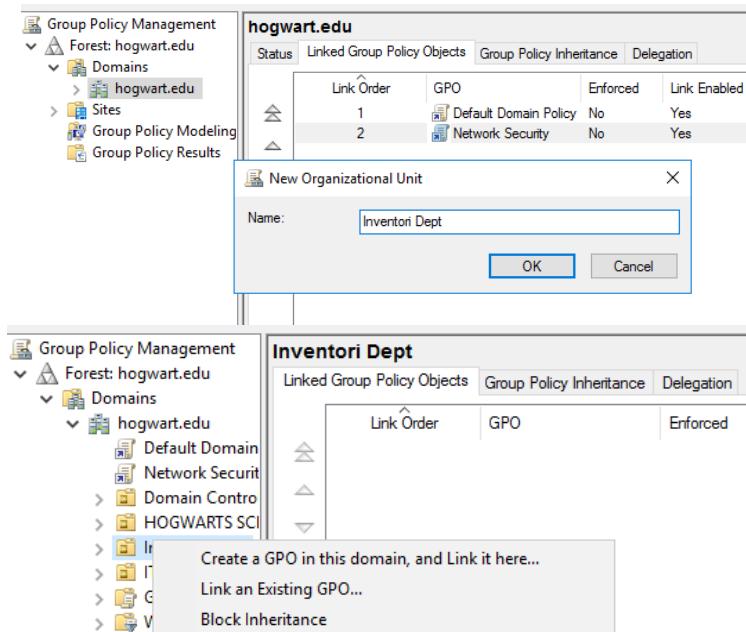


## B. Jenis-jenis Penerapan Group Policy (GP) pada OU

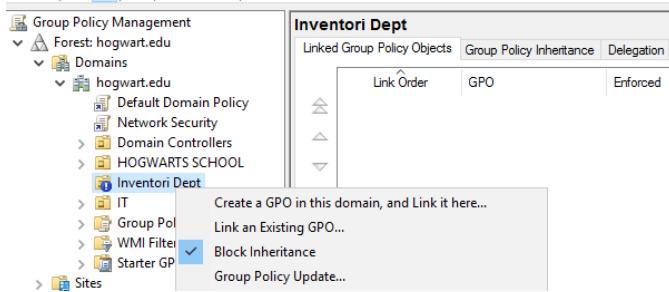
### 1. Block Inheritance GP

\*catatan: Anda harus menambahkan OU baru pada DC hogwarts.edu dengan nama Inventori Dept.

Untuk memblok pewarisan group policy (inheritance group policy) kepada sebuah Organizational Unit (OU), maka klik kanan pada OU target dan pilih pilihan Block Inheritance.



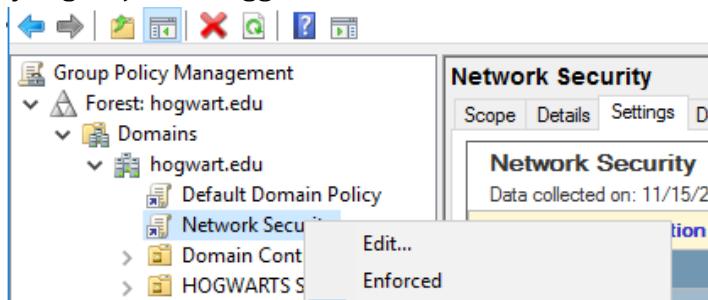
Hasilnya GP akan diblok pada OU target.



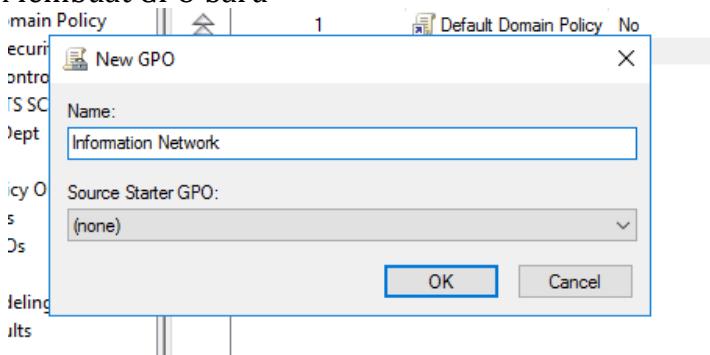
## 2. Enforced GP

Enforced GP akan memberlakukan policy kepada semua OU yang berada dibawah Active Directory. Artinya, ketika GP ditegakkan (enforce) maka OU yang telah diberikan jenis Block Inheritance GP pun akan ditegakkan untuk mengambil alih (override) policy yang telah diberlakukan. Oleh karena itu, harus berhati-hati ketika memilih jenis penerapan GP ini terutama jika sudah ditentukan bahwa ada OU yang akan menerima GP yang berbeda. Untuk mengaktifkan Enforced GP, klik kanan GPO (contoh: Network Security) lalu pilih enforced.

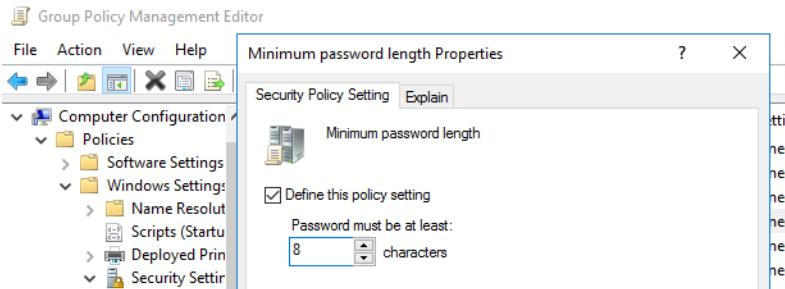
**\*catatan:** untuk membuktikan enforced GP, buatlah GP yang lain dengan nama Information Security, kemudian modifikasi dengan policy yang berbeda dari Network Security. Lalu aktifkan jenis Enforced GP pada Information security. Amati apakah policy pada OU yang telah diset pada nomor 1 berubah? Tunjukkan perubahan policy yang terjadi menggunakan cara A.6.



### Membuat GPO baru



Kemudian kita menset policy pada GPO



Kemudian kita dapat melihat setting pada direktori inventori dept

Precedence	GPO	Location	GPO Status
1 (Enforced)	Network Security	hogwart.edu	Enabled
2 (Enforced)	Information Network	hogwart.edu	Enabled

### 3. Link Enabled GP

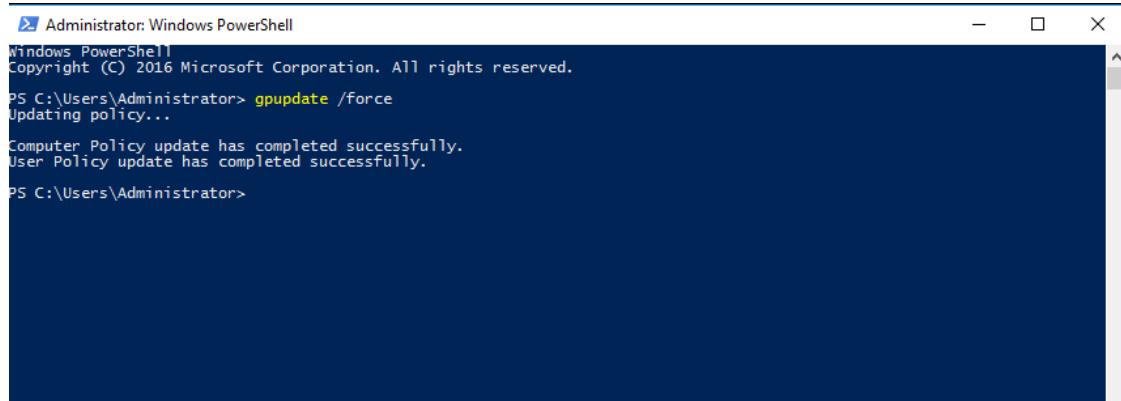
Linked Enabled GP akan memberlakukan bahwa policy akan ditarik ke OU, sehingga policy akan diterapkan kepada seluruh objek yang berada dibawah OU tersebut. Klik kanan pada GPO dan pilih Link enabled.

**\*catatan:** buatlah GPO yang baru dengan nama General Policy, kemudian pada GPO ini aktifkan Link Enabled GP.

Location	Enforced	Link Enabled	Path
hogwart.edu	No	Yes	hogwart.edu

Secara default GP akan mengambil waktu sekitar 90 menit untuk mengupdate perubahan policy pada klien. Jika perubahan perlu ditetapkan secara langsung, maka dapat dijalankan dengan perintah **gpupdate /force** yang melakukan pemeriksaan dan

pembaharuan policy.



```
Administrator: Windows PowerShell
Windows PowerShell
Copyright (C) 2016 Microsoft Corporation. All rights reserved.

PS C:\Users\Administrator> gpupdate /force
Updating policy...
Computer Policy update has completed successfully.
User Policy update has completed successfully.

PS C:\Users\Administrator>
```

### C. Pengelolaan User dan Group pada AD secara advance dengan PowerShell

\***catatan:** untuk bagian ini Anda akan menggunakan windows server 2016 yang sudah diberi nama superhero.edu, kemudian superhero.edu harus didaftarkan menjadi anggotadari (member of) domain hogwarts.edu.

#### A. Last log on time

Skrip berikut ditujukan untuk mendapatkan laporan waktu terakhir pengguna log on keAD.

```
1 ## Skrip untuk memfilter user berdasarkan waktu logon terakhir ##
2
3 $htmlformat = "<style>BODY{background-color:LightBlue;}</style>"
4 Get-ADUser -Filter * -Properties "LastLogonDate" |
5 sort-object -Property LastLogonDate -Descending | Select-Object Name,LastLogonDate |
6 ConvertTo-Html -head $htmlformat -Body "<H2>Akun AD Last Login Date</H2>" | Out-File C:\lastlogon.html
7 Invoke-Expression C:\lastlogon.html
```

Ketikkanlah skrip tersebut, jalankan. Untuk mendapatkan hasil dari skrip ini, maka Anda harus menyediakan beberapa user pada superhero.edu yang kemudian akan login ke domain hogwarts.edu.

The screenshot shows the Windows PowerShell ISE interface. The top window is titled "Administrator: Windows PowerShell ISE" and contains a script named "Untitled2.ps1". The script content is:

```
1 $htmlFormat = "<style>BODY{background-color:LightBlue;}</style>"  
2 Get-ADUser -Filter * -Properties "LastLogonDate"  
3 Sort-Object -Property LastLogonDate -Descending | Select-Object Name,LastLogonDate |  
4 ConvertTo-Html -head $htmlFormat -Body "<H2>Akun AD Last Login Date</H2>" | Out-File C:\lastlogon.html  
5 Invoke-Expression C:\lastlogon.html
```

The bottom window is titled "PS C:\Users\Administrator>" and displays the command history and the output of the script execution.

## Output:

The screenshot shows a web browser window titled "C:\lastlogon.html". The page has a light blue background and displays a table with the title "Akun AD Last Login Date". The table has two columns: "Name" and "LastLogonDate". The data is as follows:

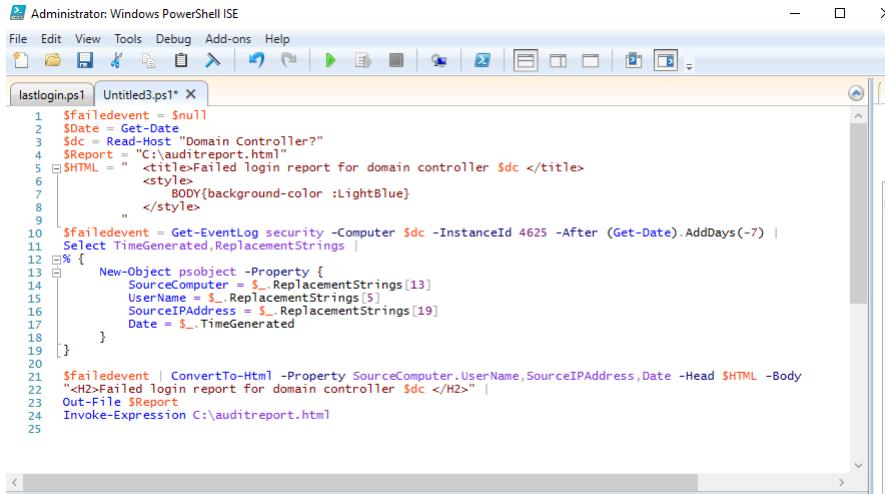
Name	LastLogonDate
Administrator	11/21/2022 5:26:32 PM
Sam	
Fox	
Meghan	
Mackie	
Anthony	
Toni Spark	
cobal	
Coba1User1	
Coba2User2	
Coba3User3	
Coba4User4	
Coba5User5	
Coba6User6	
Coba7User7	
Coba8User8	
Witwickay	
Coba9User9	
Tom	
Hermsworth	
Guest	
DefaultAccount	
krbtgt	
DOWNEY	
SCARLETT	
JOHHANSON	

## B. Failed Login Report

Skip berikut ditujukan untuk mendapatkan laporan login yang gagal ke AD.

```
1 ## laporan untuk DC Login Failures ##
2
3 $failedevent = $null
4 $Date = Get-Date
5 $dc = Read-Host "Domain Controller?"
6 $Report = "C:\auditreport.html"
7 $HTML = " <title>Failed Login report for domain controller $dc </title>
8         <style>
9             BODY{background-color :LightBlue}
10        </style>
11 "
12 $failedevent = Get-EventLog security -Computer $dc -InstanceId 4625 -After (Get-Date).AddDays(-7) |
13 Select TimeGenerated,ReplacementStrings |
14 % {
15     New-Object psobject -Property {
16         SourceComputer = $_.ReplacementStrings[13]
17         UserName = $_.ReplacementStrings[5]
18         SourceIPAddress = $_.ReplacementStrings[19]
19         Date = $_.TimeGenerated
20     }
21 }
22
23 $failedevent | ConvertTo-Html -Property SourceComputer,UserName,SourceIPAddress,Date -Head $HTML -Body
24 "<H2> Failed login report for domain controller $dc</H2>" |
25 Out-File $Report
26 Invoke-Expression C:\auditreport.html
```

Ketikkanlah skrip tersebut, jalankan. Untuk mendapatkan hasil dari skrip ini, maka Anda harus menyediakan beberapa user pada superhero.edu yang kemudian akan gagal login ke domain hogwarts.edu.

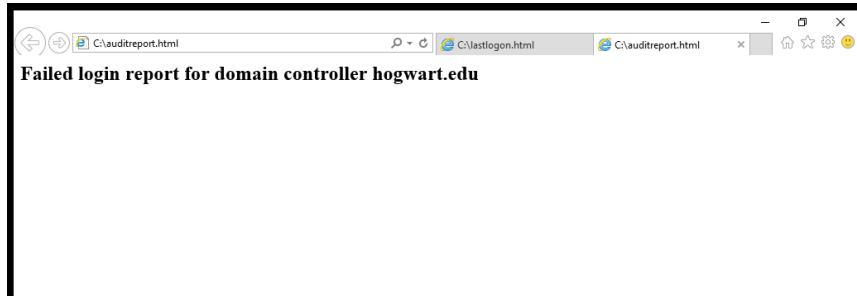


The screenshot shows the Windows PowerShell ISE interface. The title bar says "Administrator: Windows PowerShell ISE". The menu bar includes File, Edit, View, Tools, Debug, Add-ons, Help. The toolbar has icons for file operations like Open, Save, Copy, Paste, and Run. The main code editor window contains the PowerShell script "lastlogin.ps1". The script is identical to the one provided in the text above, designed to generate an HTML report of failed logins from a domain controller. The code uses PS1 syntax with variables like \$Date, \$dc, \$Report, and \$HTML, and cmdlets like Get-EventLog, New-Object, ConvertTo-Html, Out-File, and Invoke-Expression.

Output:

```
File Edit View Tools Debug Add-ons Help
lastlogin.ps1 Untitled3.ps1 X
1 $Failedevent = $Null
2 $Date = Get-Date
3 $dc = Read-Host "Domain Controller?"
4 $Report = "C:\auditreport.html"
5 $HTML = " <title>Failed login report for domain controller $dc </title>
6     <style>
7         BODY{background-color :LightBlue}
8     </style>
9 "
10 $Failedevent = Get-EventLog security -Computer $dc -InstanceId 4625 -After ($Get-Date).AddDays(-7) |
11 Select TimeGenerated,ReplacementStrings |
12 % {
13     New-Object psobject -Property {
14         SourceComputer = $_.ReplacementStrings[13]
15         UserName = $_.ReplacementStrings[5]
16         SourceIPAddress = $_.ReplacementStrings[19]
17         Date = $_.TimeGenerated
18     }
19 }
20 $Failedevent | ConvertTo-HTML -Property SourceComputer.UserName,SourceIPAddress,Date -Head $HTML -Body
21 "<H2>Failed login report for domain controller $dc </H2>" |
22 Out-File $Report
23 Invoke-Expression C:\auditreport.html
24 Domain Controller?: hogwart.edu

Running script /calibration. Press Ctrl+Break to stop. Press Ctrl+R to break into debugger.
In 91 Col 32
```



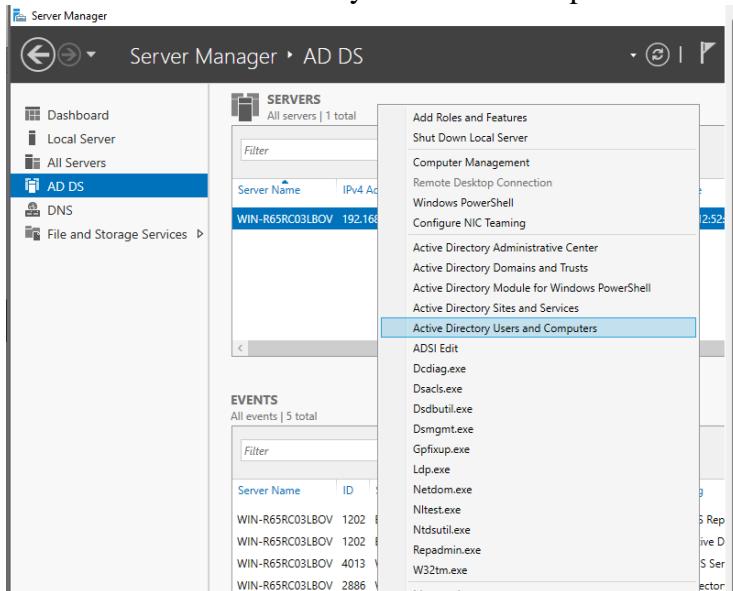
# BAB 10. AUDITING & MONITORING

## A. Langkah-langkah demonstrasi praktikum

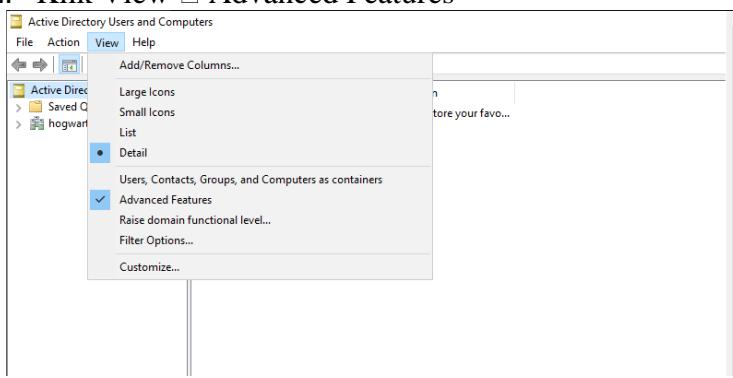
### B.1 Mengaktifkan advanced security audit policies

Ikuti langkah-langkah berikut untuk konfigurasi SACL. Setiap langkah berikan hasil screen shootnya.

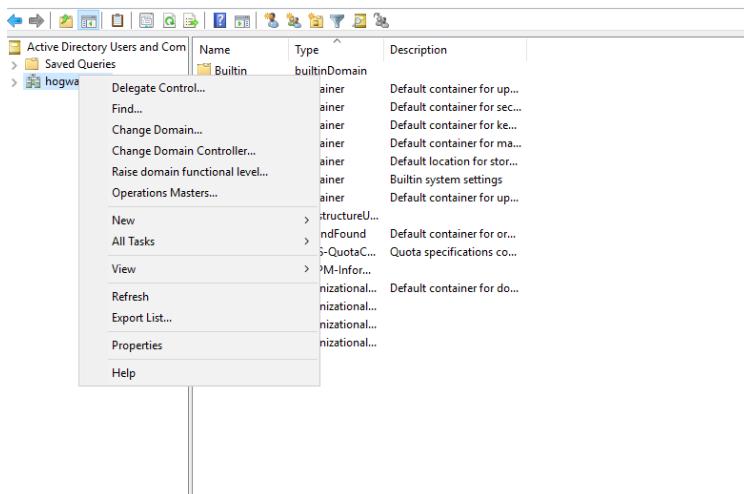
#### 1. Bukalah Active Directory Users and Computers



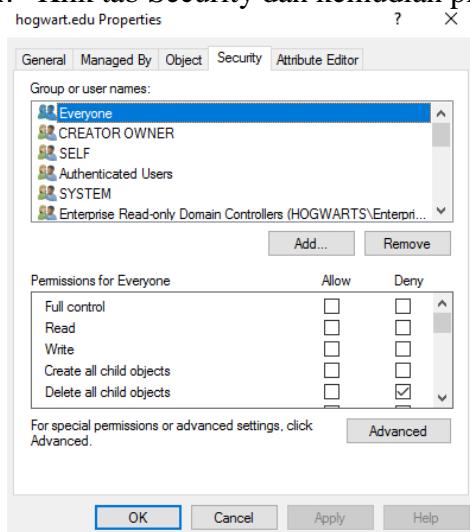
#### 2. Klik View □ Advanced Features



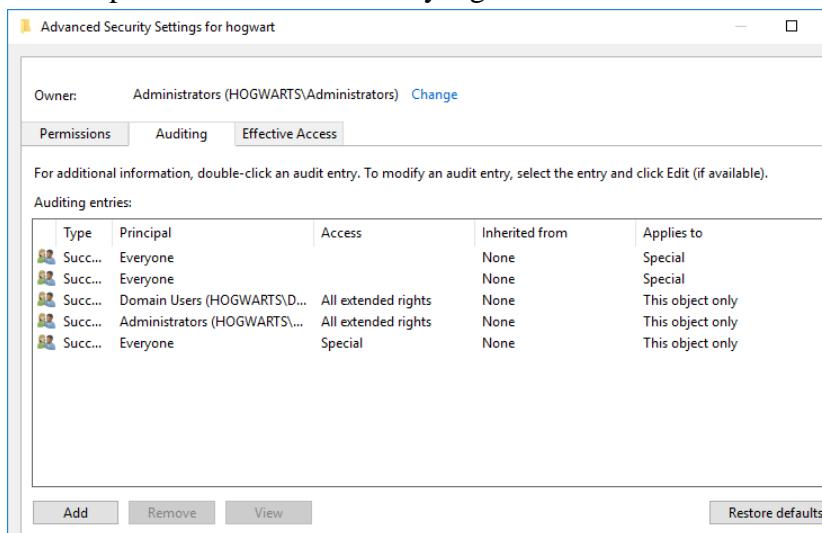
#### 3. Klik kanan pada OU atau objek yang auditingnya diaktifkan. Kemudian, klik Properties. Dalam kasus ini silahkan pilih OU IT.

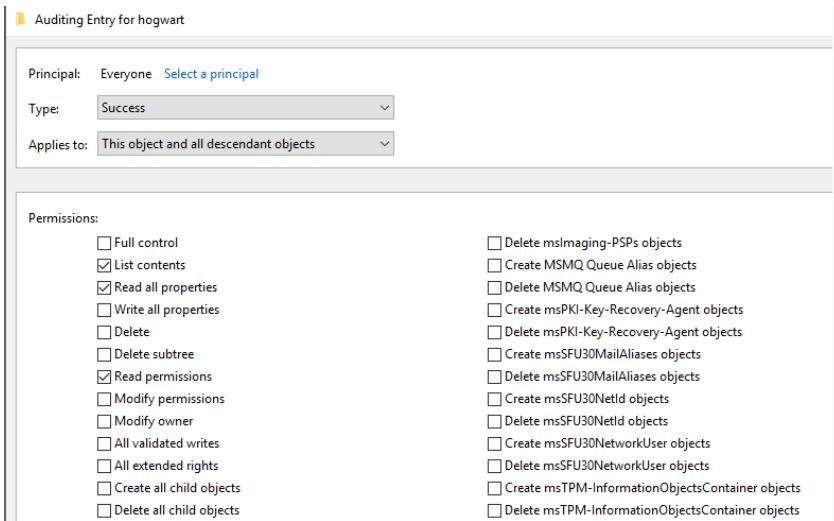


4. Klik tab Security dan kemudian pilihan Advanced

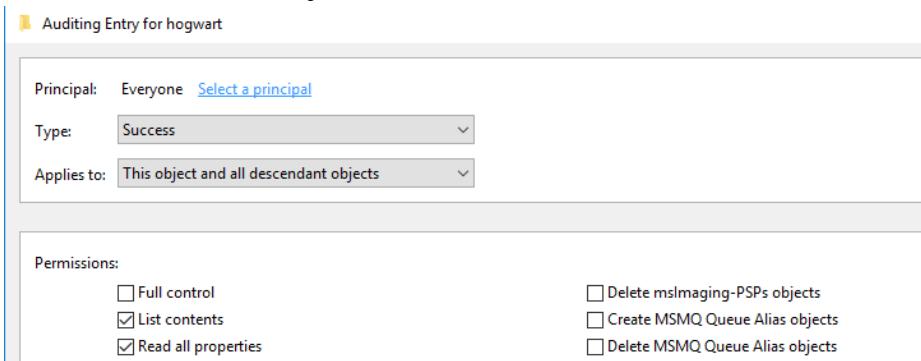


5. Klik tab Auditing, kemudian klik tombol Add untuk menambahkan aturan keamanan yang baru ke SACL. Pada skenario ini, pada security principle pilih everyone yang artinya diterapkan untuk seluruh user yang berada dibawah OU IT.





6. Untuk pilihan type pilih tipe event Success, yang kemudian diterapkan ke This object and all descendant objects.



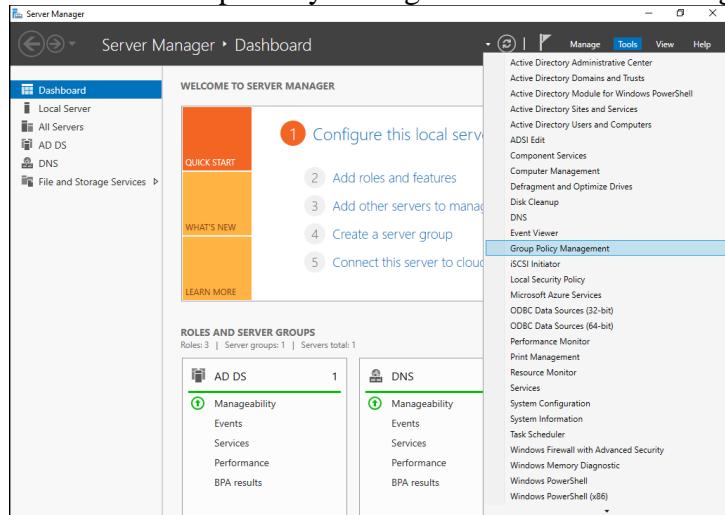
7. Pilihlah permission yang akan ditetapkan. Dari permission secara default, pilih tambahan permission lainnya seperti menciptakan user dan menghapus user.

Advanced Security Settings for hogwart				
Owner: Administrators (HOGWARTS\Administrators) <a href="#">Change</a>				
<a href="#">Permissions</a> <a href="#">Auditing</a> <a href="#">Effective Access</a>				
For additional information, double-click an audit entry. To modify an audit entry, select the entry and click Edit (if available).				
Auditing entries:				
Type	Principal	Access	Inherited from	Applies to
	Domain Users (HOGWARTS\...)	All extended rights	None	This object only
	Administrators (HOGWARTS\...)	All extended rights	None	This object only
	Everyone	Special	None	This object only
	Everyone	Special	None	This object and all descendant..
	Everyone	None	Special	Special
	Everyone	None	Special	Special

## B.2 Mengaktifkan audit policy DS Events

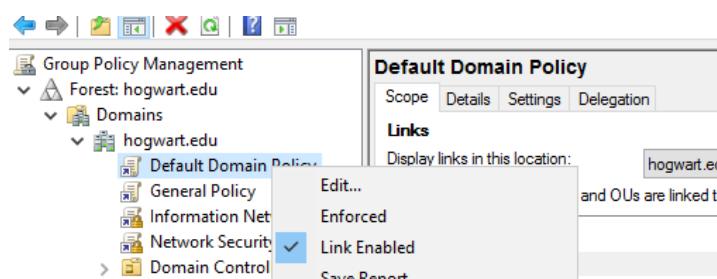
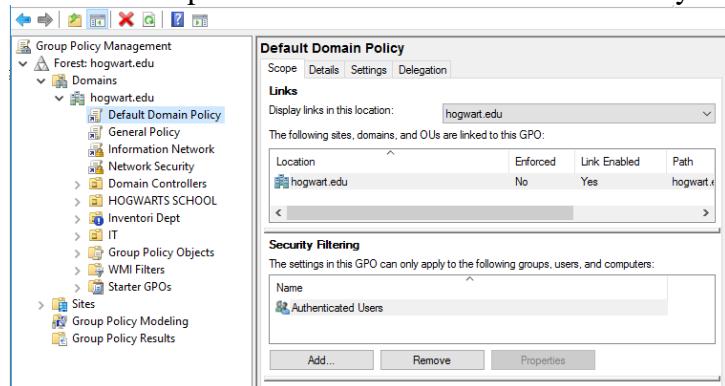
Setelah SACL diaktifkan, selanjutnya ikuti langkah-langkah berikut mengaktifkan advanced audit policy untuk DS, sebagai berikut:

**1. Bukalah Group Policy Management dari server manager.**

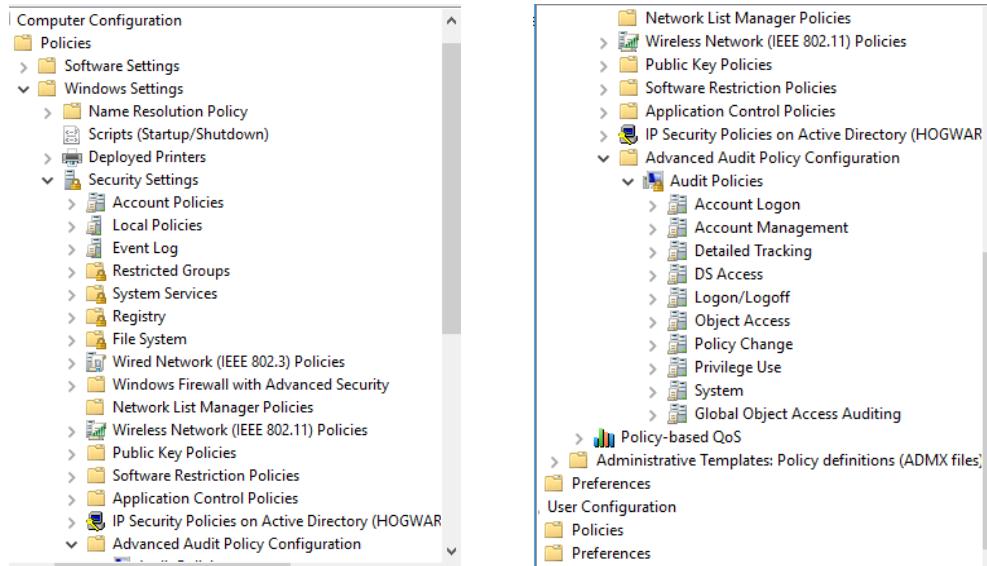


**2. Selanjutnya ekspan Domain Controller OU.**

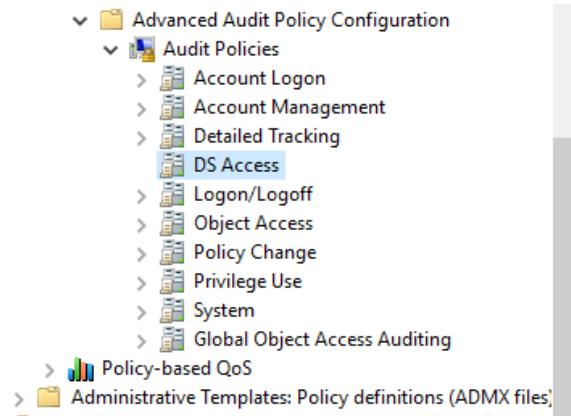
**3. Klik kanan pada Default Domain Controller Policy dan pilih Edit.**



**4. Arahkan ke Computer Configuration | Policies | Windows Settings | Security Settings | advanced Audit Policy Configuration | audit Policies,**

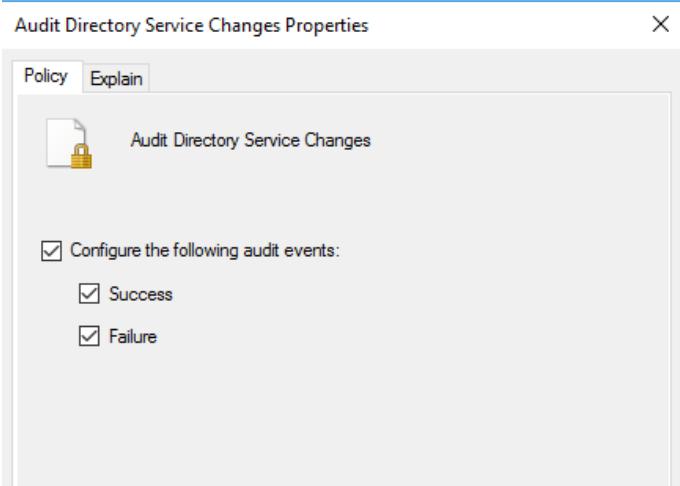


5. Kemudian, akan ditemukan 10 kategori audit. Pada demonstrasi ini kategori yang dipilih adalah DS Access dan DS changes.

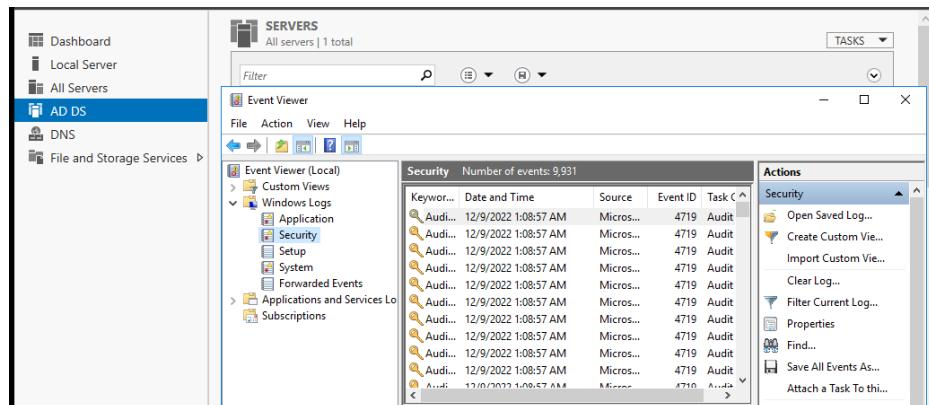
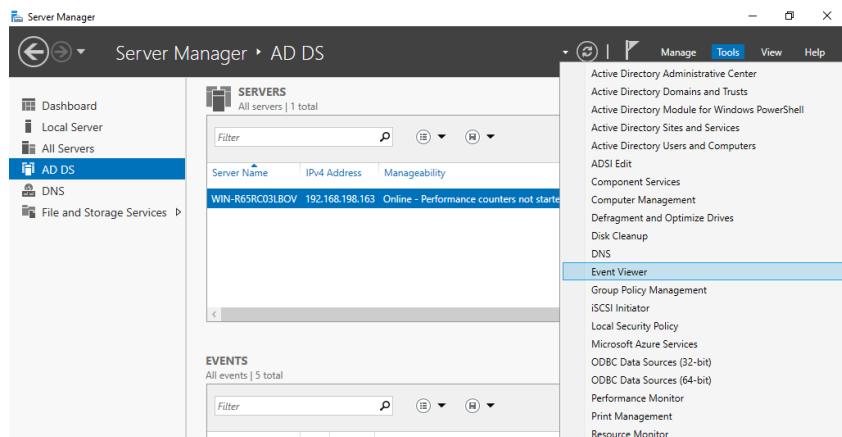


6. Arahkan ke DS Access dan double-klik ke Subcategory. Untuk mengaktifkan auditing, pilihlah Configure the following audit events dan kemudian pilih event yang akan diaudit. Direkomendasikan memilih keduanya yaitu Success dan Failure.

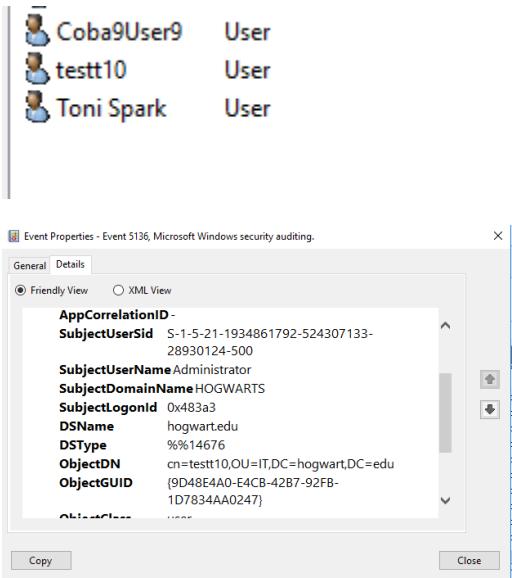
Subcategory	Audit Events
Audit Detailed Directory Service Replication	Not Configured
Audit Directory Service Access	Not Configured
<b>Audit Directory Service Changes</b>	<b>Not Configured</b>
Audit Directory Service Replication	Not Configured



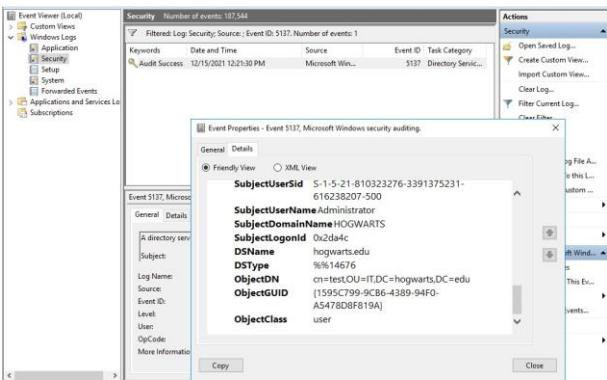
Setelah group policy berhasil diterapkan, maka events yang berelasi dengan audit policy tersebut akan dituliskan di log. Anda dapat melihatnya di Event viewer, dengan cara melakukan filter log. Lakukan uji coba dengan cara menciptakan user baru, menghapus user, kemudian periksalah event pada Event Viewer. Anda dapat memfilter event ID yang akan diperiksa dengan mengacu pada event ID di Tabel 1.



Membuat user dengan nama testt10



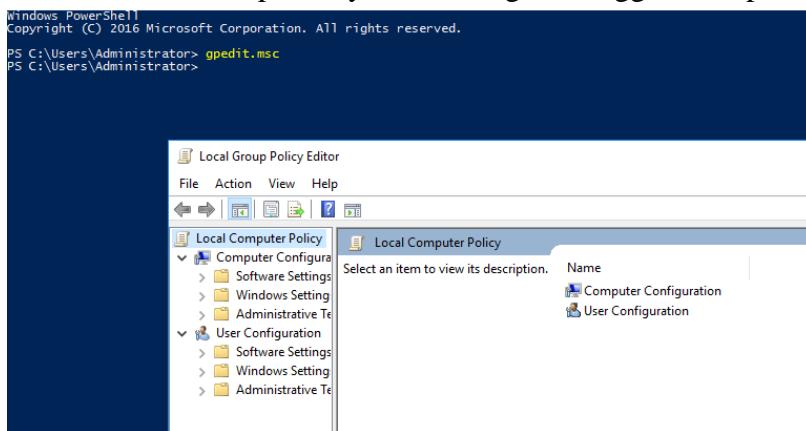
Hasilnya seperti berikut. Keterangannya adalah sebuah user dengan username test ditambahkan di OU IT oleh administrator.



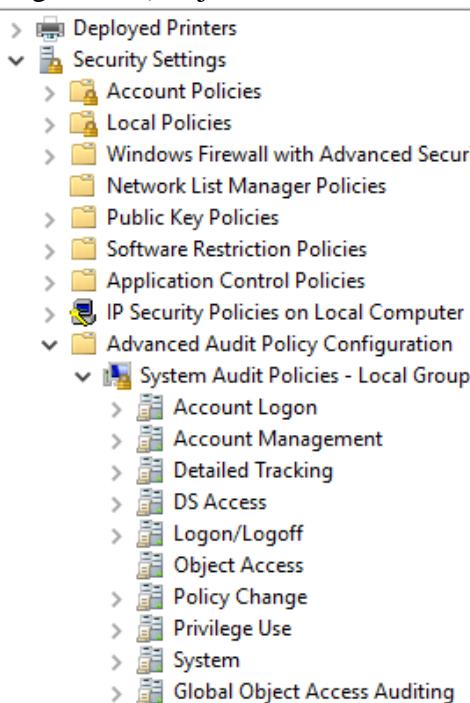
### B.3 Mengaktifkan audit policy Object Access

Pada bagian Object Access ini, akan diaktifkan auditing policy untuk akses file dan folder pada Windows Server 2016. Secara default file system object access audit policy tidak diaktifkan pada Windows server. Hal ini bisa dilakukan dengan cara mengkonfigurasi pengaturan audit menggunakan Group Policy. Pada praktikum ini akan mendemonstrasikan konfigurasi hanya pada satu komputer saja, sehingga bisa menggunakan Local Group Policy Editor. Untuk melakukan konfigurasi, ikuti langkah-langkah berikut:

1. Buka Local Group Policy Editor dengan menggunakan perintah gpedit.msc



2. Kemudian pilih Windows settings | Security Settings | Advanced Audit Policy Configuration | Object Access



3. Buka Audit File system policy kemudian pilih kedua event yang akan direkam yaitu

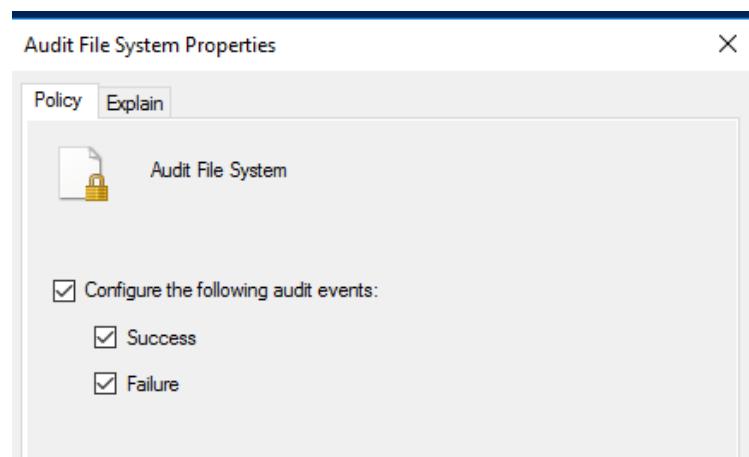
## Success dan Failure.

The screenshot shows the Windows Security Settings snap-in. Under 'Security Settings', 'Advanced Audit Policies' is expanded, and 'Object Access' is selected. A table lists various audit subcategories and their current configuration status:

Subcategory	Audit Events
Audit Application Generated	Not Configured
Audit Certification Services	Not Configured
Audit Detailed File Share	Not Configured
Audit File Share	Not Configured
Audit File System	Not Configured
Audit Filtering Platform Connection	Not Configured
Audit Filtering Platform Packet Drop	Not Configured
Audit Handle Manipulation	Not Configured
Audit Kernel Object	Not Configured
Audit Other Object Access Events	Not Configured
Audit Registry	Not Configured
Audit Removable Storage	Not Configured
Audit SAM	Not Configured
Audit Central Access Policy Staging	Not Configured

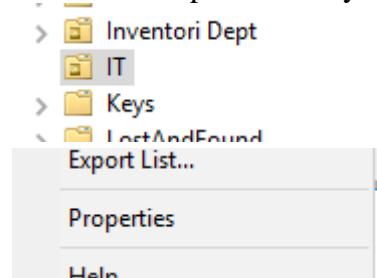
Below the table, a list of audit events is shown with their current status:

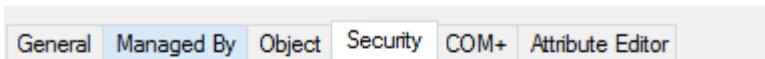
- [010] Audit Detailed File Share Not Configured
- [010] Audit File Share Not Configured
- [010] Audit File System Not Configured**
- [010] Audit Filtering Platform Connection Not Configured
- [010] Audit Filtering Platform Packet Drop Not Configured
- [010] Audit Handle Manipulation Not Configured
- [010] Audit Kernel Object Not Configured



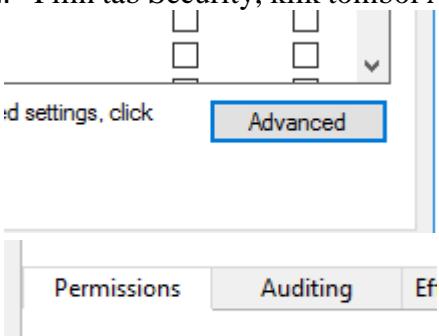
Selanjutnya lakukan konfigurasi pada folder yang akan dibagi lewat jaringan komputer, dengan langkah-langkah berikut:

1. Klik kanan pada folder yang akan dibagi, lalu pilih Properties.

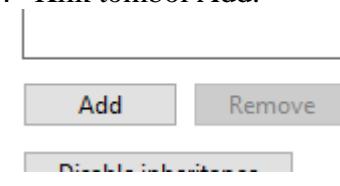




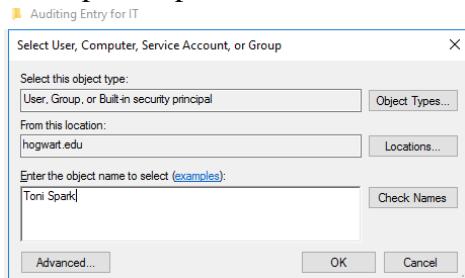
2. Pilih tab Security, klik tombol Advanced lalu pilih tab Auditing.



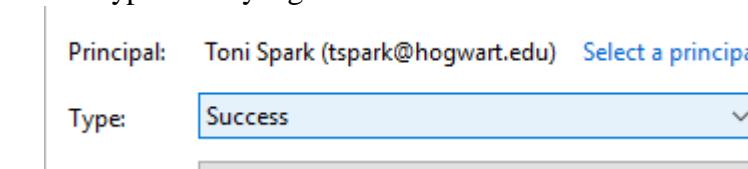
3. Klik tombol Add.



4. Pilihlah Principal dalam hal ini Anda bisa menggunakan user beyonce yang sudah diciptakan pada OU IT.



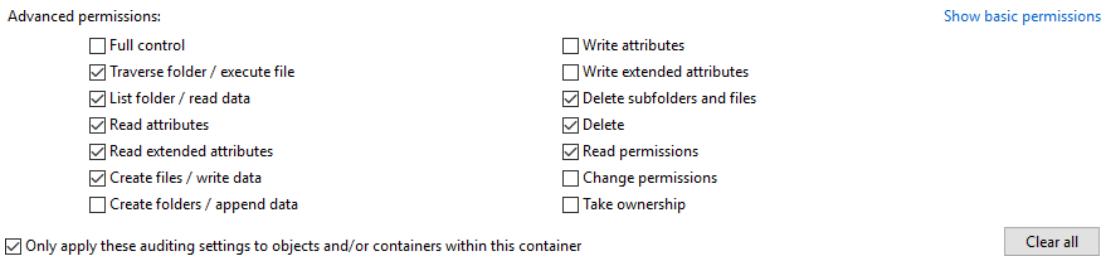
5. Pilih type event yang akan direkam adalah Success



6. Kemudain applies to, This folder, subfolders, and files.



7. Kemudian beri permission sesuai gambar berikut. Anda bisa melakukannya dengan klik Show advanced permission.



8. Kemudian klik tombol Ok, Apply, dst. untuk menerapkan konfigurasi.

Untuk uji coba, Anda harus membuat sebuah folder yang dibagi (share) dari Windows Server, kemudian pada mesin yang berbeda harus dapat diakses oleh user yang telah Anda berikan audit policy dalam kasus ini adalah user beyonce. Pada folder yang anda bagi, silahkan buat beberapa file dan hapus file tersebut. Anda bisa mencoba untuk mencari event ID 4663 ketika sebuah file dihapus oleh user beyonce. Hasil dari eksekuksi ini

Keywords	Date and Time	Source	Event ID	Task Category
Audit Success	12/15/2021 12:18:28 PM	Microsoft Win...	4663	File System
Audit Success	12/15/2021 12:08:45 PM	Microsoft Win...	4663	File System
Audit Success	12/15/2021 12:08:43 PM	Microsoft Win...	4663	File System
Audit Success	12/15/2021 12:08:38 PM	Microsoft Win...	4663	File System
Audit Success	12/15/2021 10:53:27 AM	Microsoft Win...	4663	Kernel Object

dapat dilihat pada gambar berikut.

Event Viewer (Local)

File Action View Help

Event Viewer (Local)

Custom Views

Windows Logs

- Application
- Security
- Setup
- System

Forwarded Events

Applications and Services Logs

Subscriptions

Security Number of events: 9,943 (1) New events available

Keywords	Date a...	Source	Event...	Task Category
Audit ...	12/9/2...	Microsoft Wi...	4662	Directory Servi...
Audit ...	12/9/2...	Microsoft Wi...	4662	Directory Servi...
Audit ...	12/9/2...	Microsoft Wi...	4662	Directory Servi...
Audit ...	12/9/2...	Microsoft Wi...	4662	Directory Servi...
Audit ...	11/15/...	Microsoft Wi...	4662	Directory Servi...
Audit ...	12/9/2...	Microsoft Wi...	4662	Directory Servi...
Audit ...	12/9/2...	Microsoft Wi...	4662	Directory Servi...
Audit ...	12/9/2...	Microsoft Wi...	4662	Directory Servi...
Audit ...	12/9/2...	Microsoft Wi...	4662	Directory Servi...
Audit ...	12/9/2...	Microsoft Wi...	4662	Directory Servi...
Audit ...	10/25/...	Microsoft Wi...	4672	Special Logon
Audit ...	10/25/...	Microsoft Wi...	4672	Special Logon
Audit ...	10/25/...	Microsoft Wi...	4672	Special Logon
Audit ...	10/25/...	Microsoft Wi...	4672	Special Logon
Audit ...	10/25/...	Microsoft Wi...	4672	Special Logon
Audit ...	10/25/...	Microsoft Wi...	4672	Special Logon
Audit ...	10/25/...	Microsoft Wi...	4672	Special Logon
Audit ...	12/9/2...	Microsoft Wi...	4672	Special Logon
Audit ...	10/25/...	Microsoft Wi...	4672	Special Logon

Actions

- Security
- Open Saved I
- Create Custo
- Import Custc
- Clear Log...
- Filter Current
- Properties
- Find...
- Save All Even
- Attach a Task
- View
- Refresh
- Help

Event 4662, Microsoft Windows security auditing.

General Details

An operation was performed on an object.

Subject:

Log Name:	Security
Source:	Microsoft Windows security
Event ID:	4662
Level:	Information
User:	N/A
Logged:	12/9/2022 1:08:36 AM
Task Category:	Directory Service Access
Keywords:	Audit Success
Computer:	WIN-R65RC03LBOV.hogwart.edu